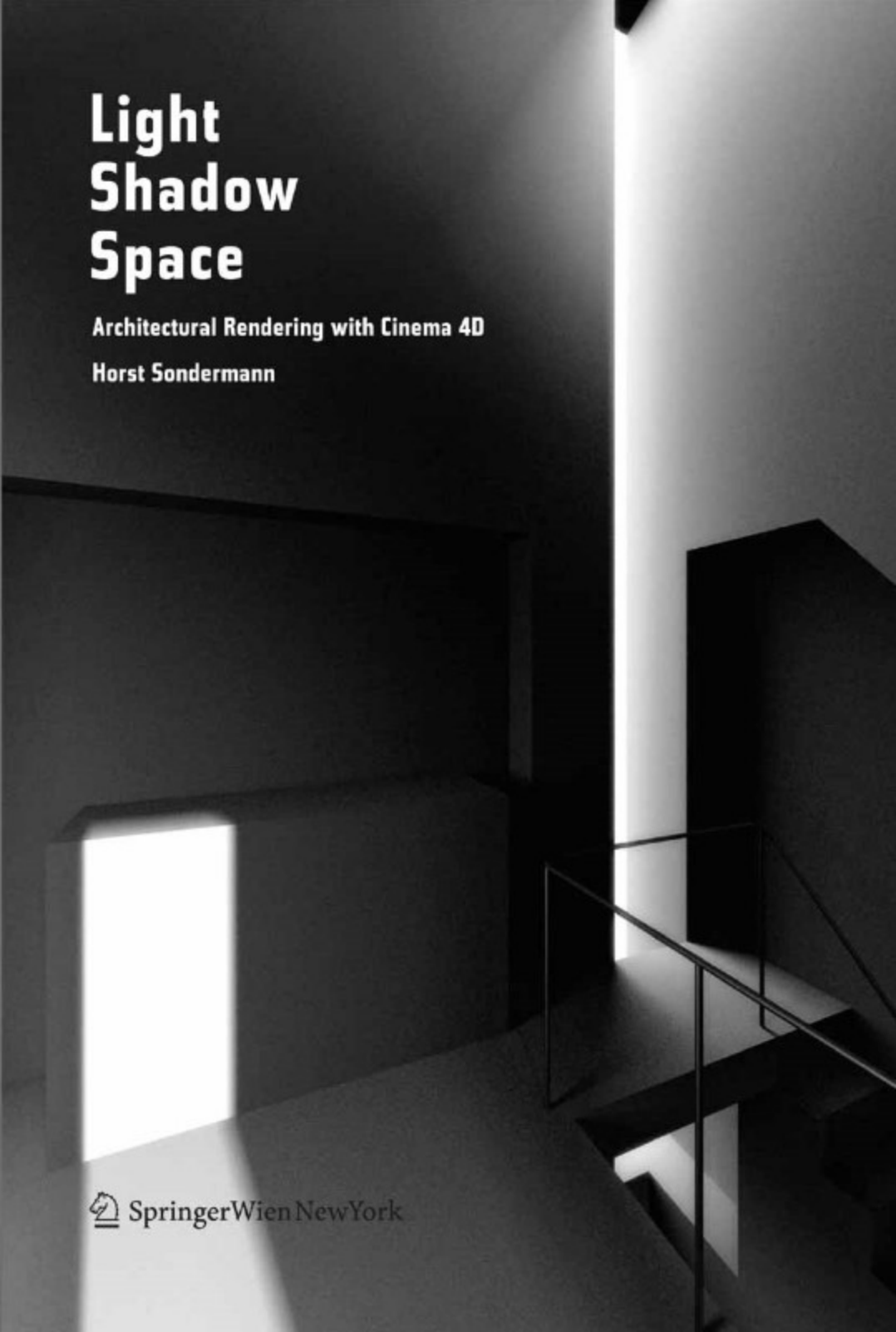


Light Shadow Space

Architectural Rendering with Cinema 4D

Horst Sondermann

SpringerWienNewYork

 SpringerWienNewYork

Horst Sondermann

Light Shadow Space · Architectural Rendering with Cinema 4D®

SpringerWienNewYork

Professor Arch. Horst Sondermann
Faculty of Architecture and Design
University of Applied Science
Schellingstr. 24, 70174 Stuttgart

Printed in Austria

SpringerWienNewYork is a part of Springer Science+Business Media
springer.at

Layout and Cover Design: Horst Sondermann
Typesetting in Tasse Regular Wide and Bold Wide
Translation into English: Pedro M. Lopez, 1050 Vienna, Austria
Printing and binding: Holzhausen Druck & Medien GmbH, 1140 Vienna, Austria

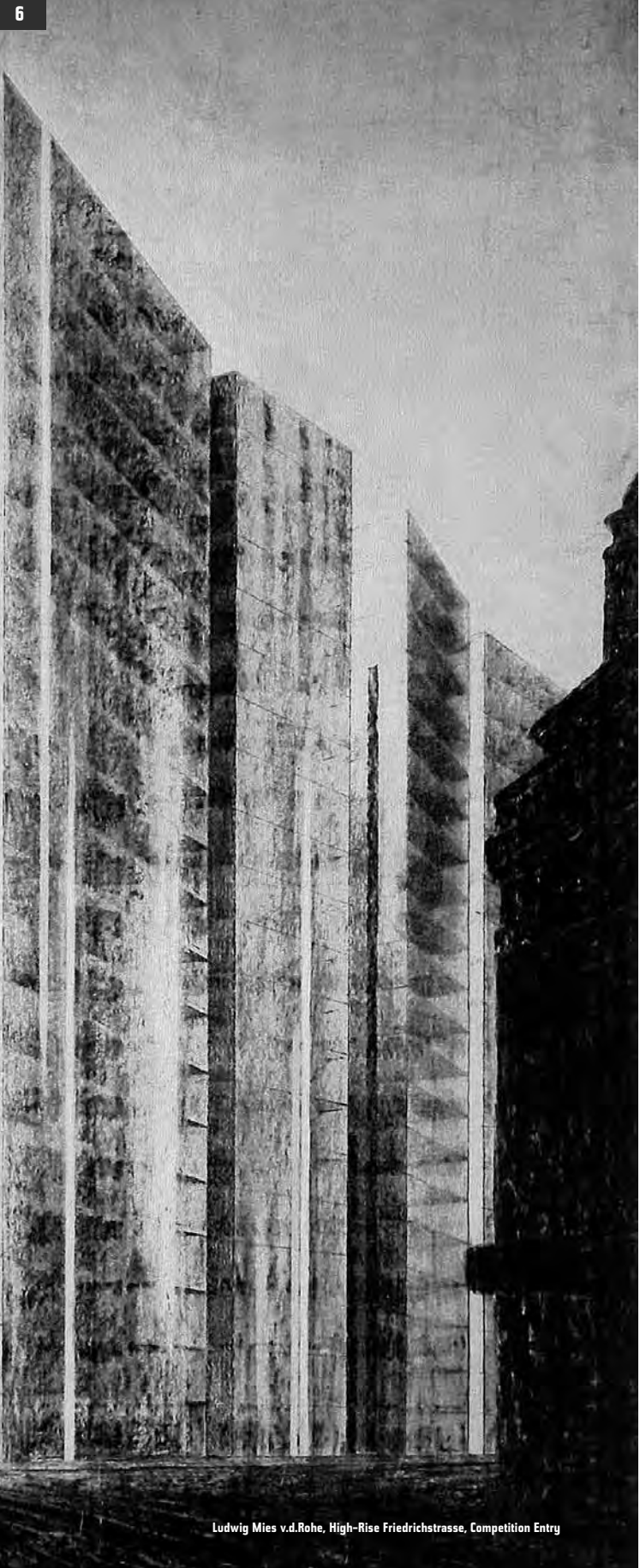
Printed on acid-free and chlorine-free bleached paper
SPIN: 11920137

CIP data applied for

With 1060 coloured Figures

ISBN 978-3-211-48761-7 SpringerWienNewYork

Table of Contents



Ludwig Mies v.d.Rohe, High-Rise Friedrichstrasse, Competition Entry

Introduction

The depiction of architecture is an old art, it plays an outstanding role in the culture of architecture – not least of all with projects that were never realized. The desire to shape the sphere of what is possible, or even the visionary aspects of a building structure on paper, canvas or on screen – the omnipresence of computers having led to an exponential increase in this desire - is the underlying motivation of this art. This makes it ever more difficult for those creating these depictions to define, develop and hone what was earlier known as a "style," to perfection. The tools change all too quickly, and it isn't easy to keep things in perspective, or differentiate between what is useful and useless.

It therefore seems daring to try to make a helpful contribution to orientation in the confusing field of digital architectural rendering with a slim volume such as this. But there are, in brief, two important reasons for which I nonetheless made an attempt.

Firstly, I would find it a shame to not be able to use the potential of 3D software for one's own artistic purposes – major architectural aspects can be visualized with it all too well: space, color, materiality, light and shadow. The two last aspects are of special importance in my view since they make the use of 3D software most justifiable, while the other aspects could as well be displayed with currently available CAAD software.

The second reason for my work is the desire to separate what is practical and manageable from what is technically possible - in my office and teaching experience, architectural rendering in everyday planning is often a matter of a few days or hours. The technology involved has to meet these requirements, or it is useless otherwise.

I chose to use Cinema 4D by Maxon to address my first reason – not least because of the free language layouts it comes with. Additionally, it offers a broad spectrum of interfaces for both easy access to CAAD-generated material as well as several export formats for post-production purposes.

The desire to filter what is useful in everyday life results in a deliberate limitation of the contents; it might be baffling that I do not discuss the use of Radiosity (or Global Illumination) in relation to the most important subjects, light and shadow. But in my opinion, the additive, "lacquering," i.e. layering of individual lights with ray tracing is the more "craftsman-like" technique, which allows for easier control (esp. of the rendering time). Hence the exclusion of Global Illumination seems justified to me, especially since the new light sources available in Cinema 4D´s 10 version already offers very far-reaching realistic light visualization possibilities. Animation was also excluded – except for a detail in Chapter 3. In my opinion, it is a specialized field that should be discussed in a separate book.

In fact, the book concentrates on creating lighting sets for digital architecture models, which I supply, including prominent examples such as the new National Gallery in Berlin built by Ludwig Miesv.d.Rohe, or the remodeled Tomba Mambretti by Guiseppe Terragni, the interior of which is featured on the cover.

The subject is discussed in general introductory chapters on Cinema 4D, as well as those on handling polygon bodies, importing from and those on light and shadow types in Cinema 4D. In closing, the book includes a chapter on modeling, texturing and multi-pass rendering. The creation of lighting models is described step by step to make every step verifiable – this made the repetition of certain things unavoidable. I nonetheless chose this approach to enable the reader to work through each chapter as an independent unit without having to leaf through the book searching for references.

In some places it was important to me to show the post-processing possibilities Photoshop, Adobe's image-editing classic, offers. This is the case in chapter 16, Multi-Pass Rendering and Compositing, which would be incomplete without a reference to post-production possibilities.

The files needed to work on the individual chapters are also available as downloads on the Internet, please read the details at the end of the book.

Other important details you should make a note of are: the command key is referred to as the ctrl key (on a Mac, which I also use, it is the cmd key), the select key is referred to as alt (or opt key on certain Macs). I also assume that you use a mouse with more than one key. Measurements are always written in abstract units (without metric designations), since Cinema handles things the same way.

Light Shadow Space was first published in German spring 2006. The English edition, naturally, demanded illustrations being exchanged, which proved to be a good opportunity to present Cinema 4D´s revised interface in release 10. Accordingly, I felt obliged to rewrite increments of text too – which I hope gives this book enough actuality until the next major update.

With these things in mind, I hope you enjoy your reading.

01

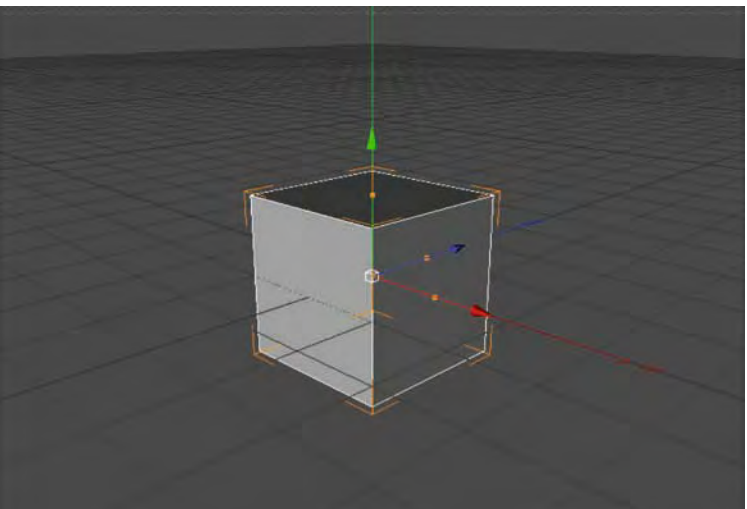
01

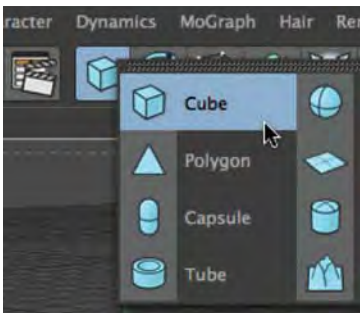
Cinema 4D · Overview and Navigation

In this chapter, I would like to make you familiar with Cinema 4D handling principles – a somewhat bold undertaking if one takes the scope of functions the program offers into account.

However, it seems sensible to me to choose certain fundamental aspects and present them to you to arouse your curiosity. Then you can explore the details in between with the help of the handbook, which I highly recommend for consultation purposes.

02





03

A Cinema 4D file looks like the image above on your screen (Illus. 01) – a large window in which you can see a model, surrounded by a number of smaller windows that seem to show details of the elements in the scene.

This type of file is referred to as a scene in 3D jargon, the large window is the viewport or Editor in which the scene is edited.

The other windows are known as managers, which display varying information on the scene and/or settings – they are equivalent to the panels you know from other programs.

When you start Cinema 4D the first thing you see is an empty scene. It is striking that each window has its own menu bar and that the windows are "stuck" to each other. You can change the size of two windows by sliding the line between them with your mouse cursor. This makes one larger and the other smaller.

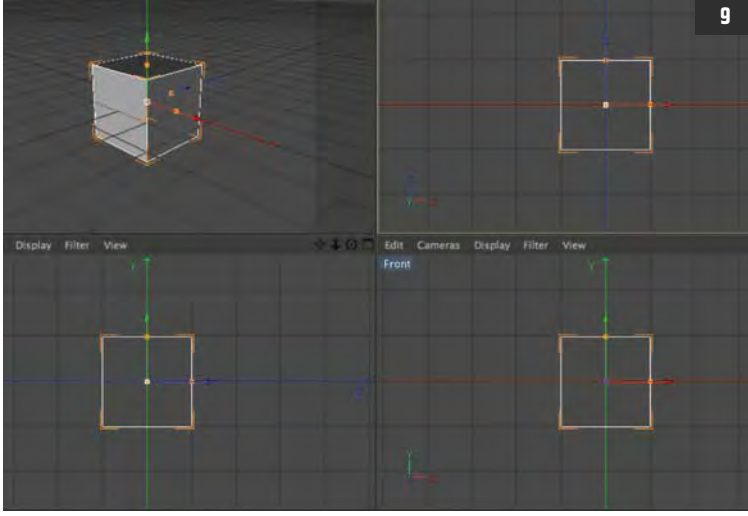
Click on the button of the Primitive Objects menu (Illus. 03) and select the cube – it is placed in the center of your scene (Illus. 02). It is displayed in shaded perspective in the Editor. The visible axes show that the cube is selected for possible editing.

Click into the Editor viewport using the scroll wheel of your mouse (middle mouse key). Now you see four viewport windows which show the various views of the cube (Illus. 04) – the perspective image is shaded, the other images merely show lines. This is a default setting that can of course be changed (we will discuss that later).

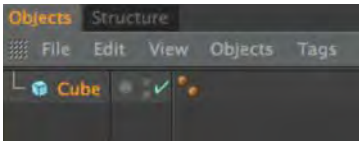
If you click into one of the four windows with your scroll wheel again, only the clicked window will be shown (e.g. the window on the upper right that shows a top view).

A separate projection can be chosen for this and every other one of the four windows in the Editor Camera menu (Illus. 06). As can be seen in the four-window image, the default settings include a perspective, top, front and side view. (You can also select these four windows directly by pressing the F1-F4 keys, the four-window image can be set by pressing F5).

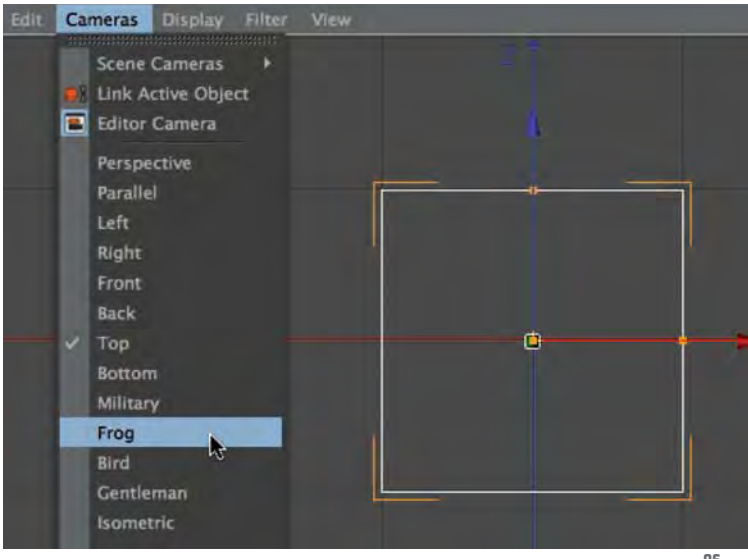
The way the views are organized can be set in the Editor View menu (Illus. 07). But use the default setting to begin with. The cube also appears in the Object Manager (Illus. 05).



04

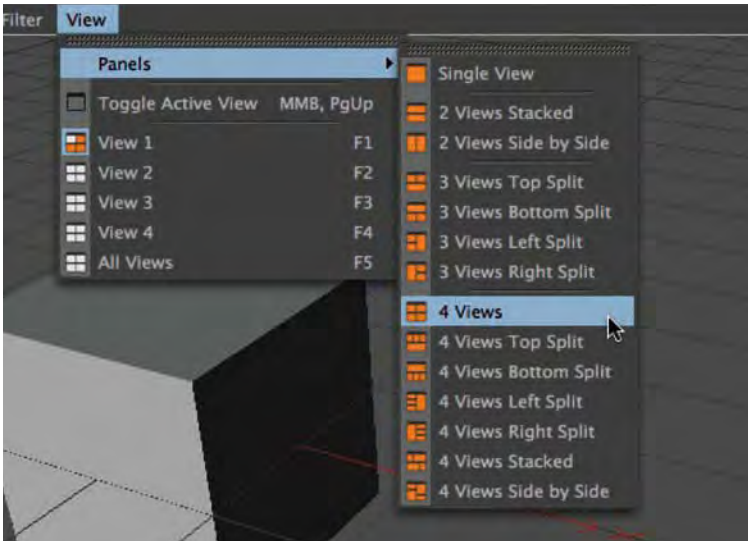


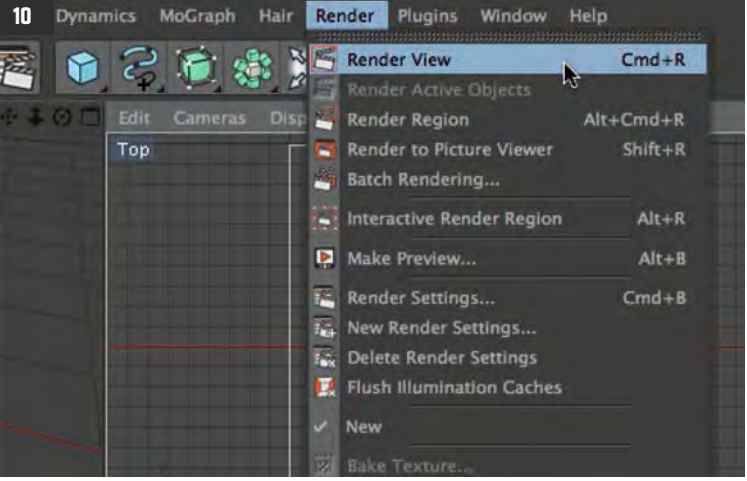
05



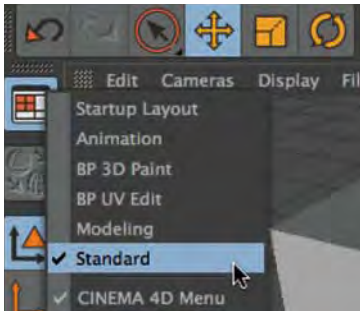
07

06





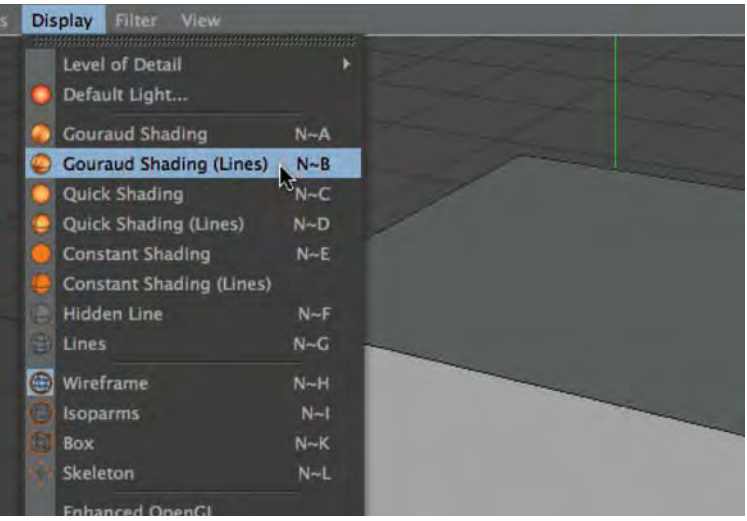
08



09

If more views are displayed, you notice that one of them has blue edges. That is the view which would be used for rendering (Render menu: Render View, Illus. 08). Any window can be selected for rendering by clicking into it.

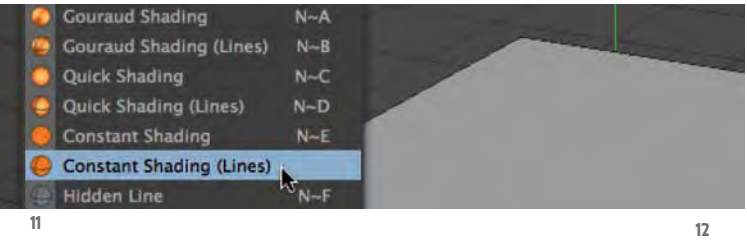
If the program's interface has become confusing after sliding the windows or hiding various managers, click on the layout button at the upper end of the command bar on the left side and choose the Standard option from the pop-down menu.
Everything should then appear the way it did when you started the program – except for the things you see in the Editor (Illus. 09). You probably already guess that you can create various surfaces yourself and have them available at this point, but that isn't an issue yet.



10

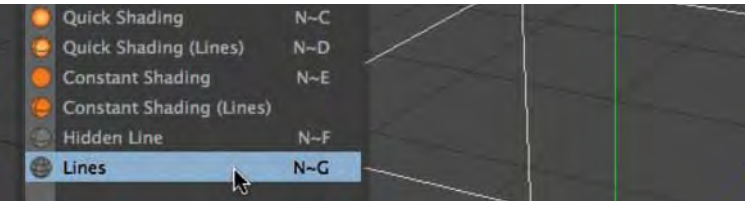
We saw earlier that the perspective view was shaded, and that the other views were only shown in line mode.
The type of depiction can be set individually for each viewport window using the Display menu in the respective viewport. You can see the display alternatives in the second field from above – Gouraud Shading, Quick-Shading etc. – and in the field below you have the options for line display (Wireframe, Isoparms, etc.).

It is best to keep the line depiction set on Wireframe, but feel free to try the shading alternatives (Illus. 10-12).
Gouraud Shading takes the location of the object surfaces into account with regard to the lighting, either the Default Light in the more or less empty scene, or with regard to existing lighting sources.
Constant Shading doesn't react to light directions at all, instead, it shows all same-color surfaces with the same brightness. The lines show the polygon edges.
The viewport windows can show differing



11

12



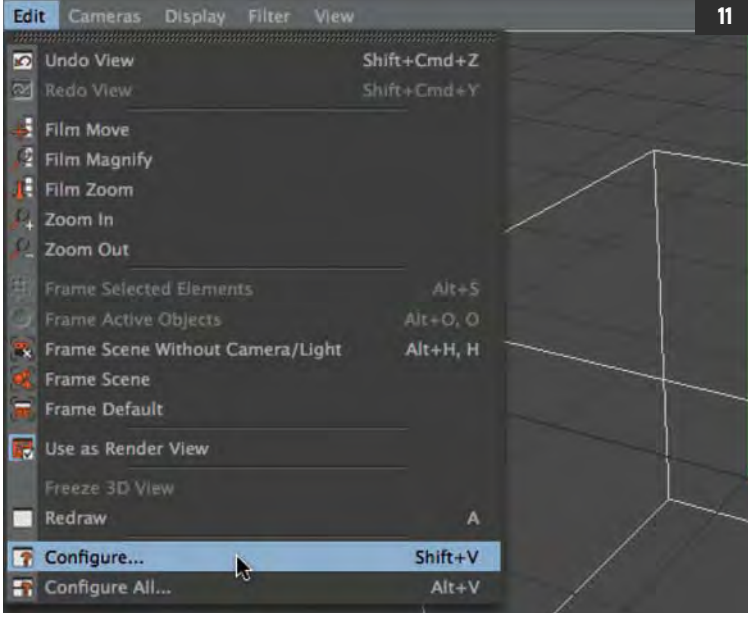
views, but you can also use one viewport window to display selected and non-selected items in different ways.

Make sure your current view shows a perspective. It is best to hide the other viewport windows (F1). Select the Configure Command (as you can see, you can also choose default settings for all viewport windows, Illus. 13).

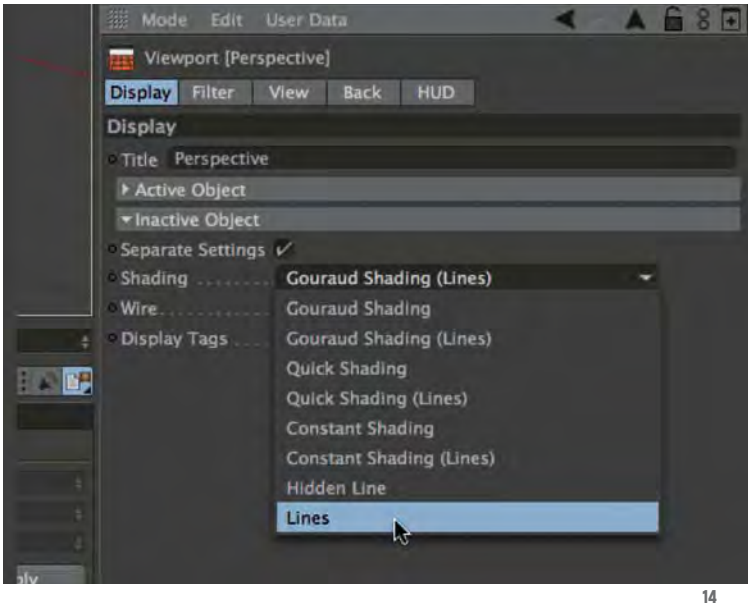
You can now see the available default windows in the Attributes Manager window on the lower right, sorted into a number of panels (Display, Filter, etc., Illus. 14).

You can choose settings for active and inactive objects in the Display panel. In the Inactive Object field, check the Separate Settings option and select the Lines option from the Shading menu (you may eventually have to click on the small triangle symbol near Inactive Object to access the settings.

Now, place a sphere selected from the Primitive Objects menu in the scene – it is shown as an active object, the cube – now inactive – is displayed in wireframe model (Illus. 15).

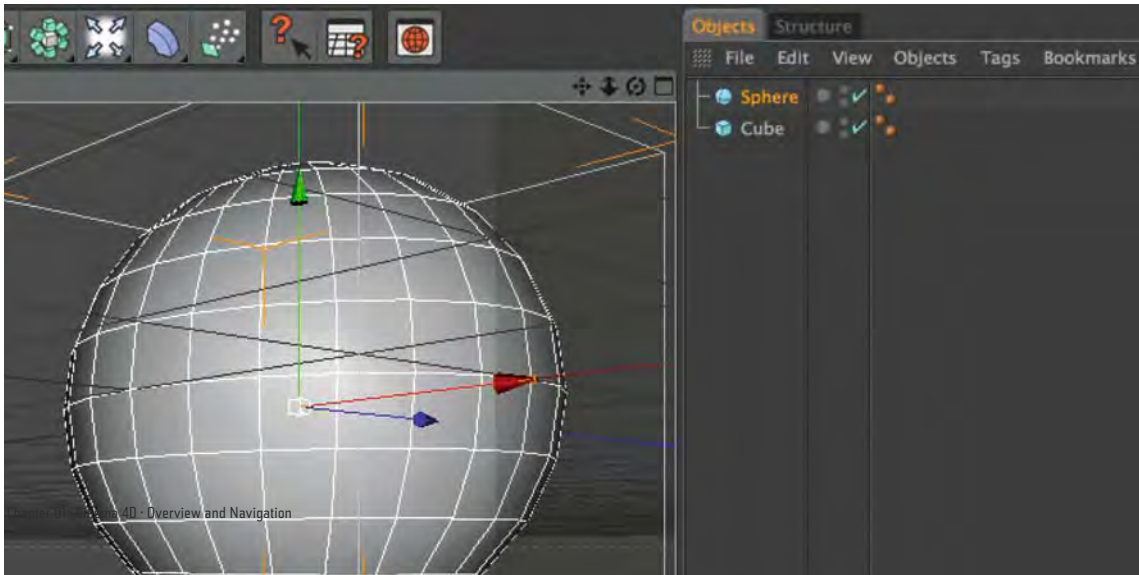


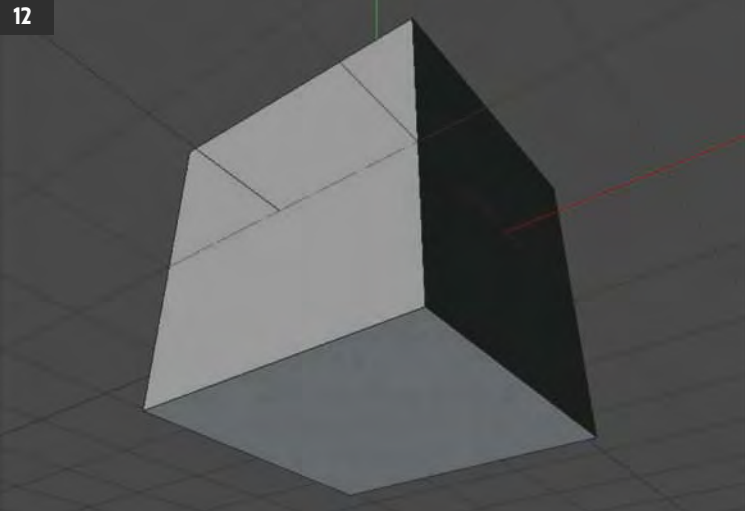
13



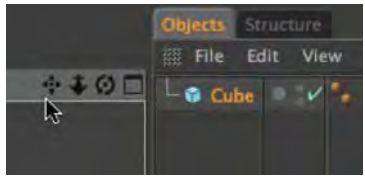
14

15





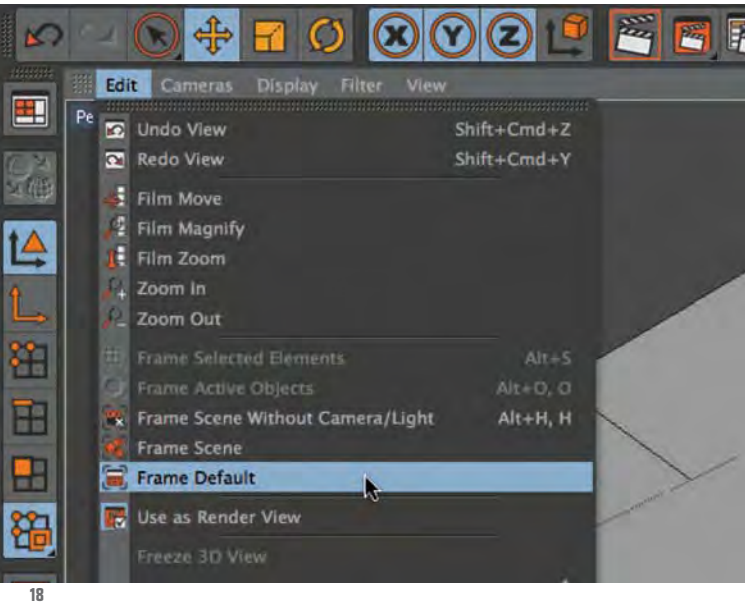
17



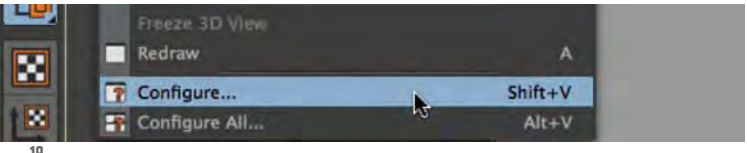
16

Now that you are familiar with the Editor display basics, it is time for you to learn how to move within your scene.

There are three buttons in the first place, which can be seen along the upper edge of the Editor window (Illus. 16; the fourth button on the far right can be used to switch between viewport windows).
One after another, please click on each of the three left buttons and keep your left mouse button pressed. Move the mouse – as you can see, the first button moves the view, the second helps you zoom in or out and the third allows you to rotate within the scene (for zooming, move your mouse left or right).



18



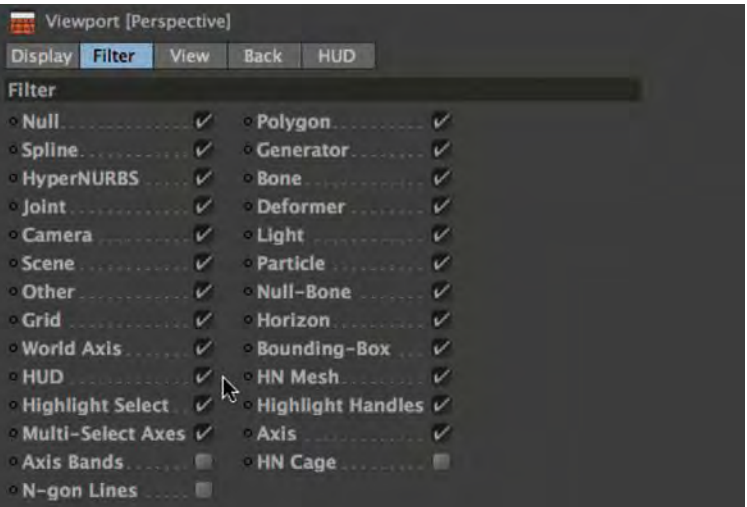
19

20

These actions do not change a thing in your scene itself, but the direction and distance of the view you take (Illus. 17). If you have lost your sense of orientation while navigating through the scene, simply select the Frame Default command from the Editor Editing menu – this resets the image to the default view (Illus. 18).

Of course a number of options you should know are concealed under the simple navigation tool surfaces – begin by imagining that you are viewing the scene through a camera (it is secondary whether these are "real" cameras you placed or the constantly present Editor camera – this subject is discussed later).

If you use the middle navigation button, you can change the size of the image – either by moving the camera or by changing the focal distance. To understand what



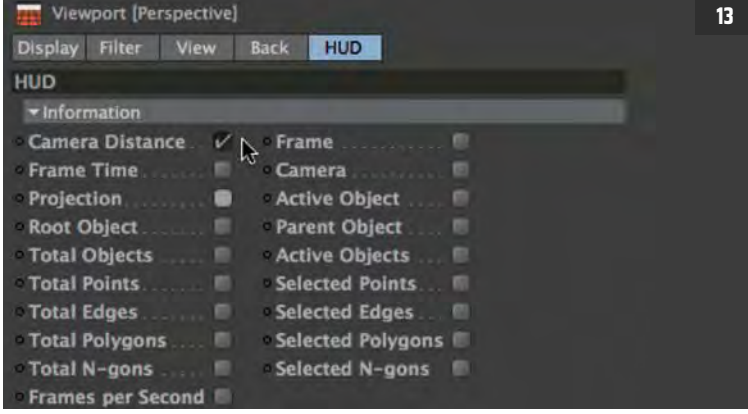
Cinema 4D does in this case, we should use a control tool, the so-called HUD (Head Up Display).
Choose the Configure settings from the Edit menu again (Illus. 19), and change to the Filter panel in the Attributes Manager. Here you can see which elements may be shown in the scene – the HUD is among them (Illus. 20). The Head Up Display can be used to show several scene and object parameters in the Editor.

Now change to the HUD panel. Here you can select the parameters to be shown – check the Camera Distance option to display the distance between the active object and the camera (Illus. 21).
Please keep the cube in your sample scene selected and reset the view with the Default Frame command (Viewport menu. The HUD shows a camera distance of 900 (Illus. 22).

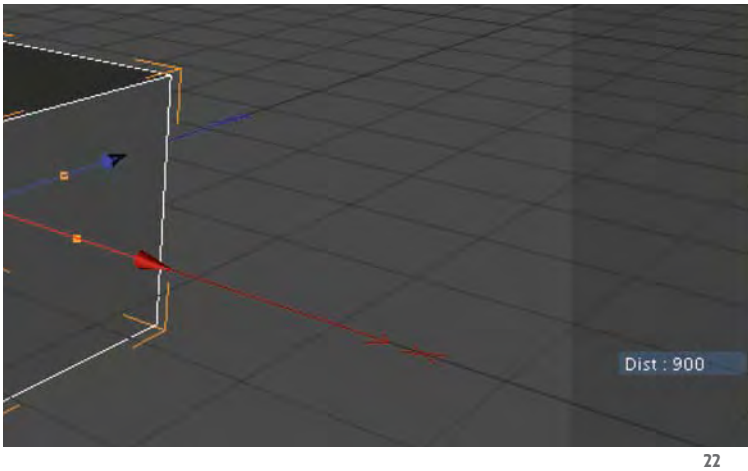
Now zoom into the scene by clicking the middle navigation button and keeping the left mouse button pressed while moving the mouse to the right. The HUD shows how the camera moves towards the cube while the distance decreases (Illus. 23).

Select the Default Frame command again and zoom out of the scene, now using the right mouse button while clicking the navigation button – as you can see, the cube becomes smaller and smaller, but the camera distance remains constant (Illus. 24).
In this case you are not moving the camera away from the object but reducing the focal distance. Your viewing angle becomes wider, and the more you see of your scene in the Editor, the smaller the size of the individual objects will become.

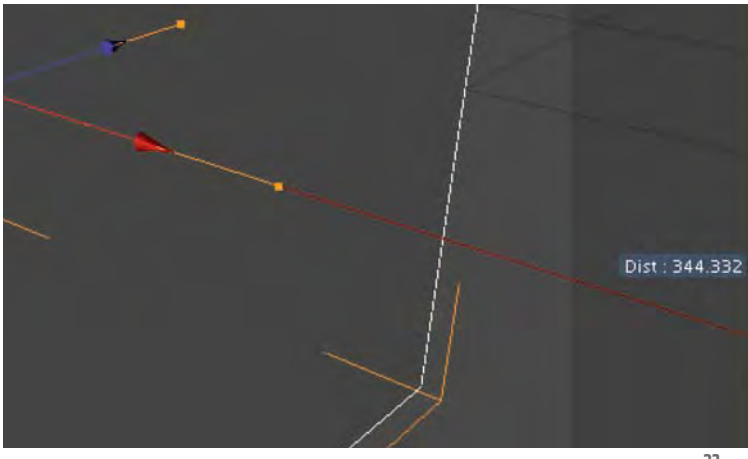
These two possible ways of scaling the image are the same ones you may be familiar with in photography.



21

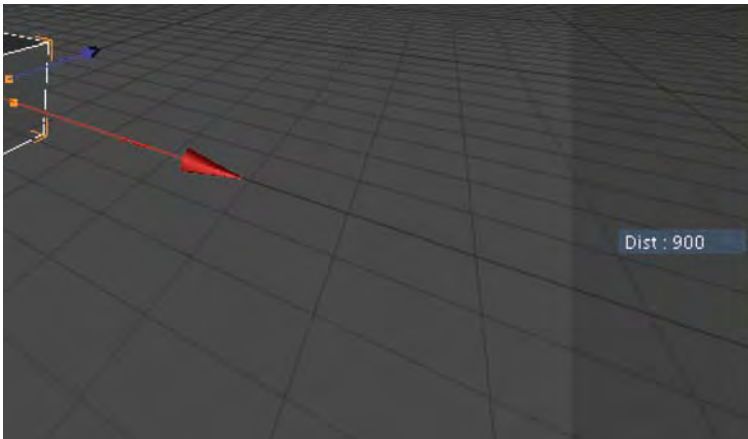


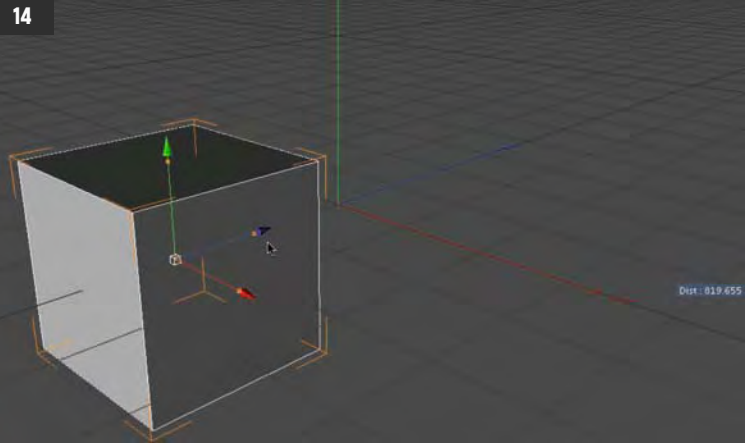
22



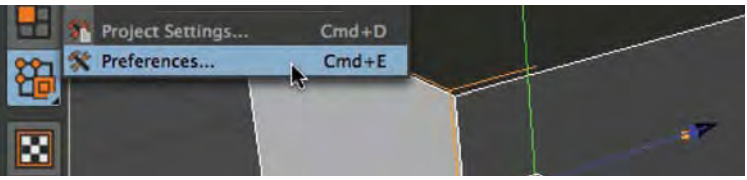
24

23

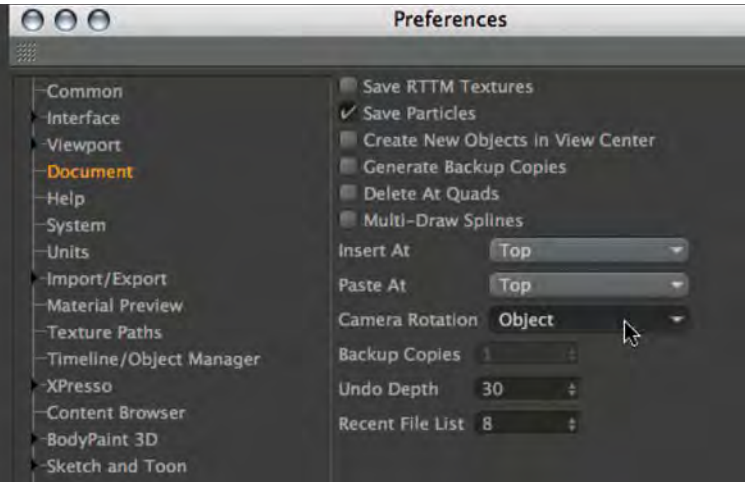




25

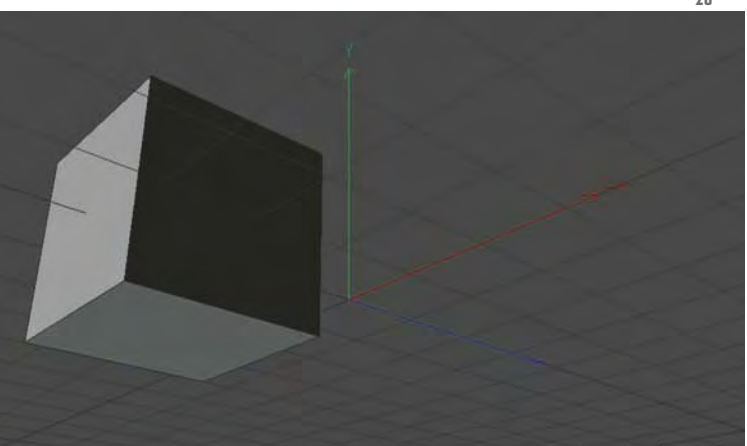


26



27

28



An alternative to the use of the navigation toggle on the upper edge of the Editor is the possibility of moving the scene with the help of the Alt key.
If you keep it pressed, you can rotate your camera with the left mouse button, shift the view with the middle mouse key (most common, it's the scrolling wheel) and zoom the camera in and out of the scene with the right key (as if you were using the middle Navigation button via the left mouse key).

Rotating the camera this way offers a number of options. To get acquainted, with them, move the cube out of the center of the scene; select it, click on the blue coordinate axis, and move the mouse to the left with the key pressed (Illus. 25).
If you turn your view now, you are rotating a virtual camera – as mentioned above. You can define which center the camera rotates around – the active object, the center of the viewport, the camera position or the point of origin of the entire scene.
The corresponding options can be selected in the program Preferences to be found in the Edit menu (under Document, Illus. 26). If, for Camera Rotation, the Object option is selected (Illus. 27), the camera will rotate around the zero point of a selected object – or around the common zero point of a number of selected objects.
If no object is active, the camera will rotate around the zero point of the scene (Illus. 28).
But that isn't all you can do – if you press the shift key as you turn, you will rotate around the center of the viewport, If you press the Ctrl instead, the camera will rotate around its own center.

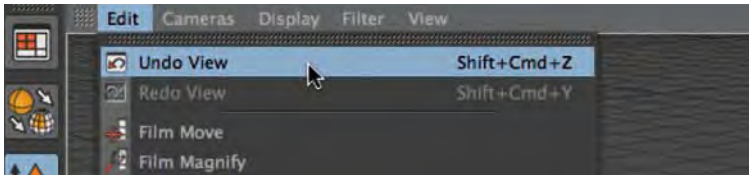
As you can see, you are provided with several camera rotation options with only one option in the program preferences selected.

After zooming to and fro, you probably want to get an overall scene view again, so select Frame Scene Without Camera/Light from the Edit menu, or simply press the H key. Cinema 4D then fits your entire scene into your window (the Alt H short cut does the same for all your open viewport windows, Illus. 29).
Other helpful functions when working in a virtual space are the Undo View and Frame Default commands in the same menu (Illus. 30).

You have learned how to control your scene display using the Editor, and how to move inside your 3D world.
It is time to learn how to handle "real" cameras, which enable you to lock your view and thus get better control of your work's output.
If your model has been constructed in another CAAD Program, you might already have one of those cameras. You could then import them using Cinema 4D's 3ds format. However, let us for now assume you want to create a camera within your Cinema 4D scene.
To do this, use the Editor to navigate towards your favored camera position.
Then pick a camera from the Scene Objects menu (Illus. 31). You won't see a change in the Editor - Cinema 4D places the camera in the current Editor camera's position using the Editor camera display settings.

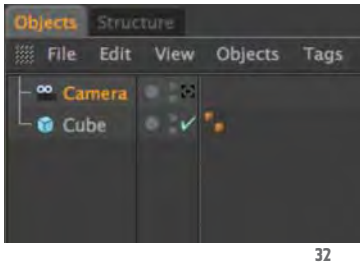
The new camera is listed in the Object Manager (Illus. 32), but it is still the Editor camera which is active. To refine the settings of your new camera, please select it by using the Scene Cameras command (Editor Cameras menu Illus. 33).
In the Object Manager, it is now marked in yellow (Illus. 34). The Attributes Manager reveals the camera parameters (Illus. 35).

29

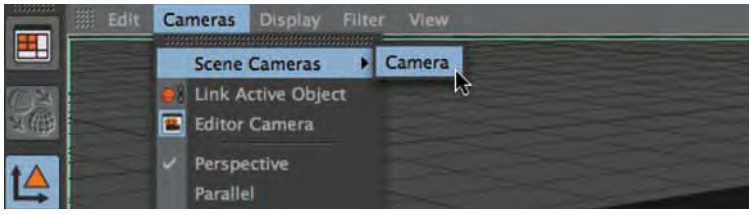


30

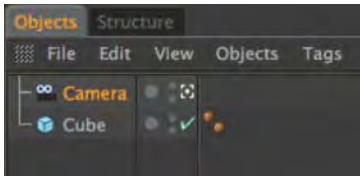
31



32

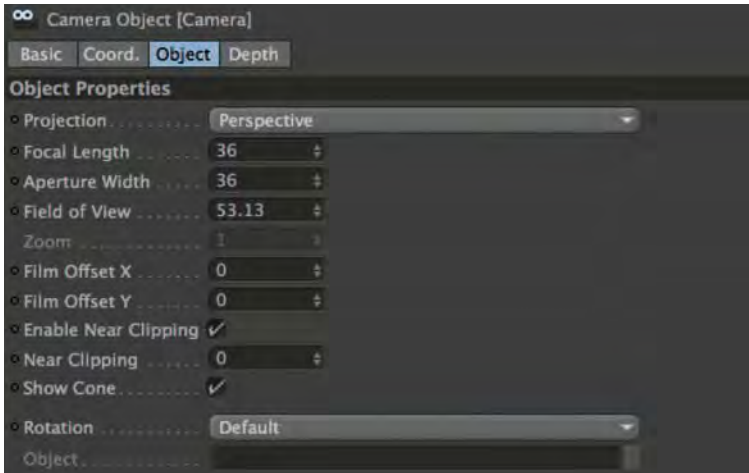


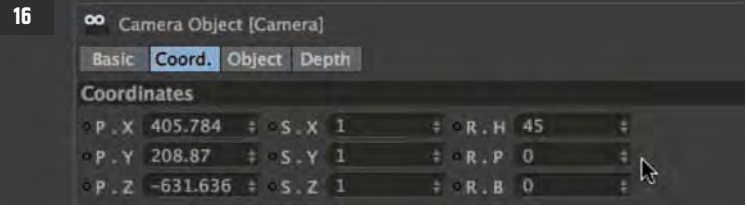
33



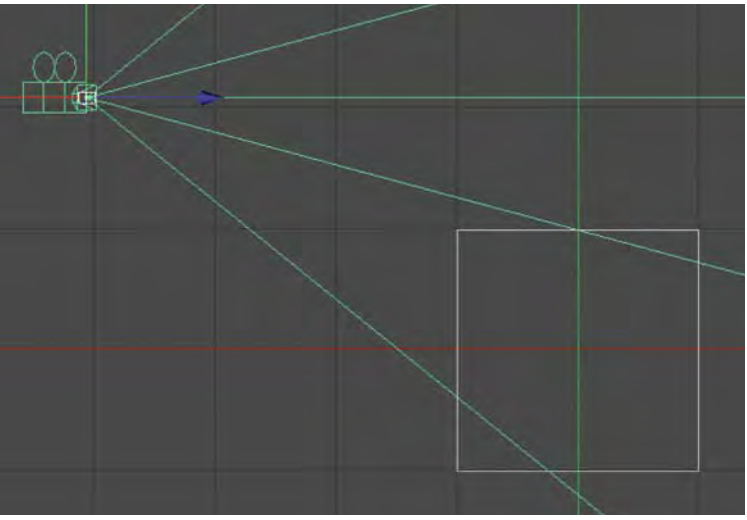
34

35

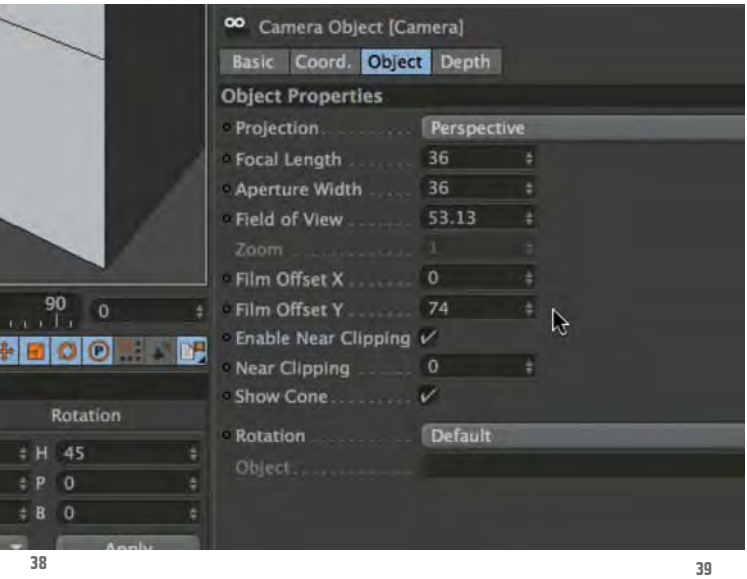




36

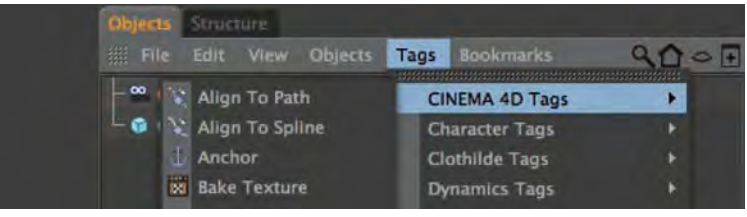


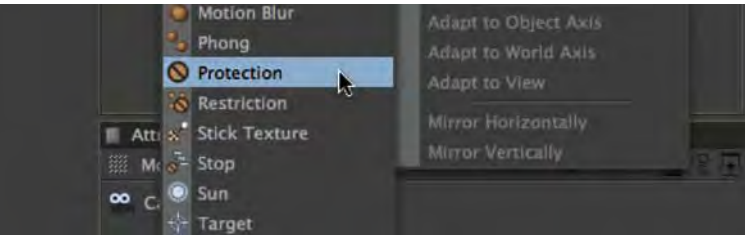
37



38

39





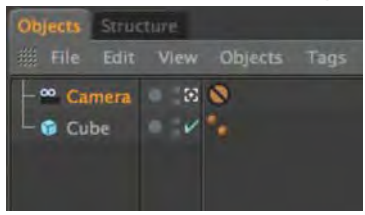
Here you can optimize the geometry of your perspective. In most cases, a good architectural image requires your model's verticals to stay perpendicular in view. To reach this, it is compulsory that your camera is precisely adjusted horizontally. (There are exceptions, of course – dramatically converging verticals can be fashionable in a scene dominated by sky scrapers, for example, or interiors.)

Have your camera selected and switch to the Coordinates panel in the Attributes Manager. Set the R.P and R.B. angles to 0° – the resulting perspective and the view windows show you that the vista axis is now horizontal (Illus. 36 and 37).

If part of the scene is no longer visible after leveling the camera, Cinema 4D offers you to shift the camera perspective vertically and horizontally. Keep your camera selected and switch to the Object panel in the Attributes Manager. Change the values for Film Offset X and Film Offset Y here. If you click on the double arrow next to the values and move your mouse while keeping its button pressed down, you can immediately see the effects in the Editor (Illus. 38). Using the offset only moves the image inside the viewport frame, the perspective's geometry remains the same.

When you're satisfied with your image's clipping, lock the camera by assigning a Protection tag (Object Manager Tags menu: Cinema 4D Tags, Protection, Illus. 39). You can identify the tag by the little icon beside the camera object in the Object Manager (Illus. 40). Now the camera is locked against changes in position and

40



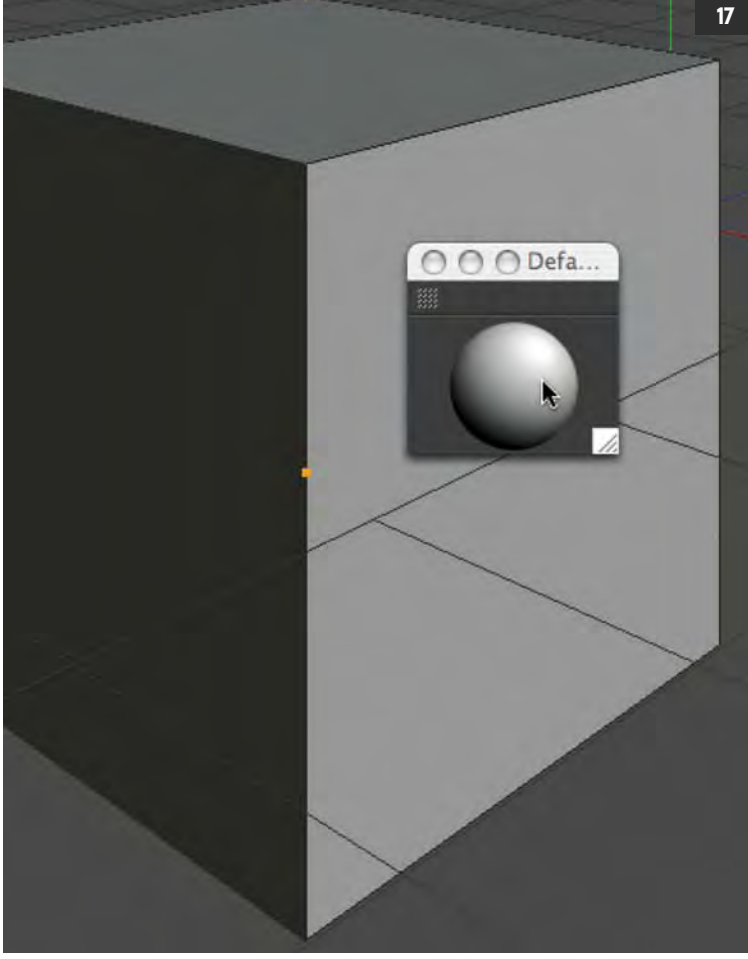
41

adjustment – Focal Length and Film Offset may still be changed.

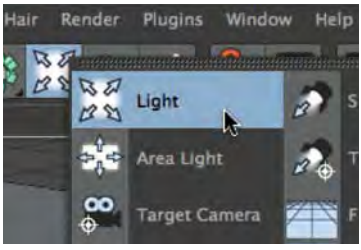
Cinema 4D allows you to illuminate a scene with a variety of virtual light sources. This is of course a vast subject, of which some aspects are described in the following chapters. Having discussed display, navigation and the installation of cameras, I would like to at least give you a foretaste of this subject.

Up to now, there are no "real" light sources in your scene, but you can recognize the 3d objects, even when rendering it. Things are visible because Cinema 4D uses a so-called Default Light that provides sufficient lighting for each new scene – until we create our own lighting setup. Default Light is deactivated as soon as we install at least one light source.

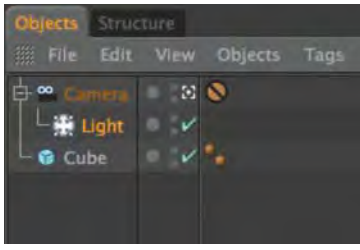
Select the Default Light command from the Editor menu Display (Illus. 41) – you can change the Default Light position by clicking on the ball that appears, a right-click on the ball will reset the parameters, Illus. 42).



42



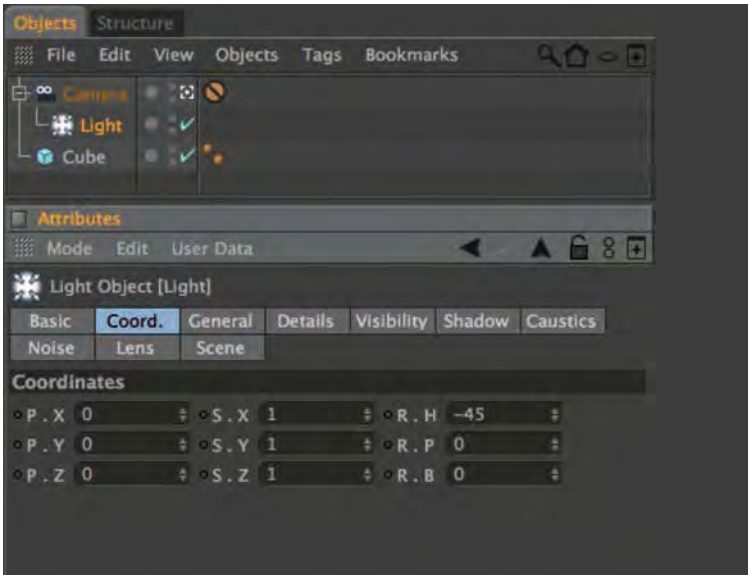
45

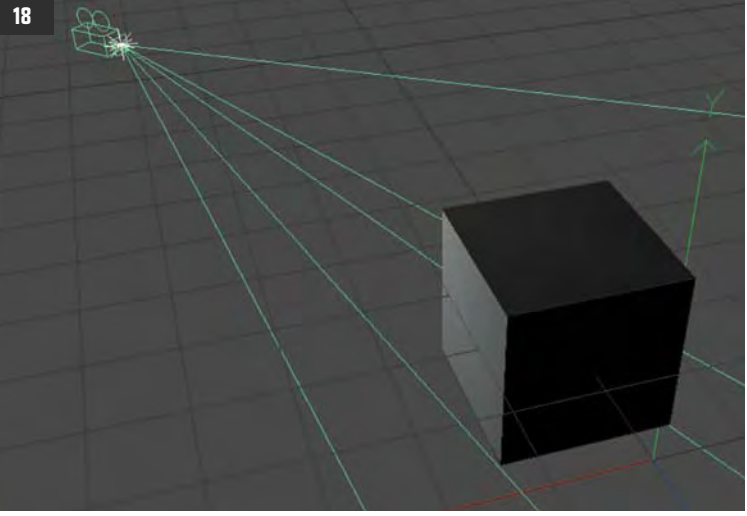


43

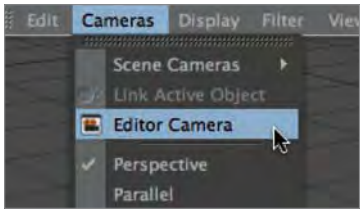
44

Now install a „real" light source in the scene (pick a Light from the Scene Objects menu in the upper command bar, Illus. 43). Drag the new light object onto the camera object in the Object Manager – this way the light becomes a sub-object to the camera (their hierarchical relation being shown in the Object Manager, Illus. 44). Have the light source selected and set the position coordinates P.X, P.Y and P.Z to 0 in the Coordinates panel of the Attributes Manager (Illus. 45).





46



47

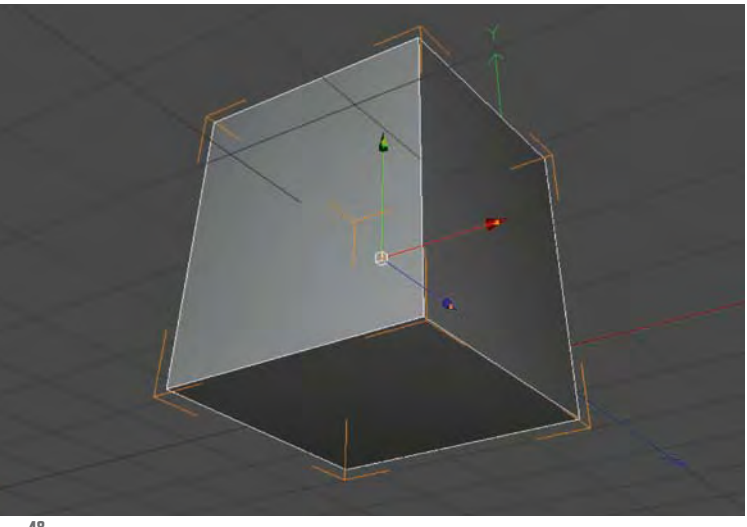
Switch to the Editor camera view again to see your new camera in its position, and the camera light attached to it (Illus. 46 and 47).

By dragging the light object onto the camera, its location is linked to the camera's – the Attributes Manager now shows its position coordinates related to the camera position, not to the scene's zero point. The X, Y and Z coordinates being zero mean that the position of camera and light is identical.

By default, the light source you installed is a so-called Omni that emits light in all directions and does not produce cast shadows.

Because of its position, everything is lit that can be seen through the camera. Such a setup is often necessary to give a scene additional lighting (or Fill, to put it more professionally).

Switch back to your camera, delete the Protection tag and rotate around the cube – as you can see, your new camera light follows every move, since it is linked to the camera (Illus. 48).



48

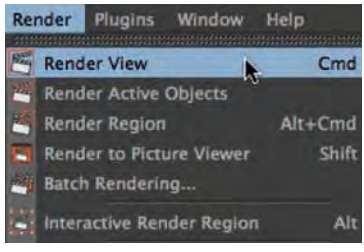
In line with the light subject, I'd like to briefly show you how to render an image in Cinema 4D, and how your objects may behave when doing so.

Everything you see in the viewport may also be rendered – just choose the Render View command from the Render menu (Illus. 50).

Cinema 4D now renders the complete image. For calculating the rendering, it takes into account every installed light, applied

49

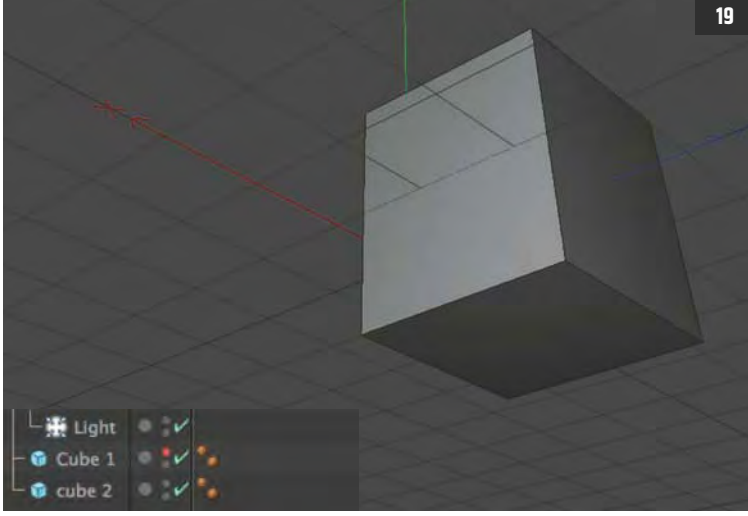
50



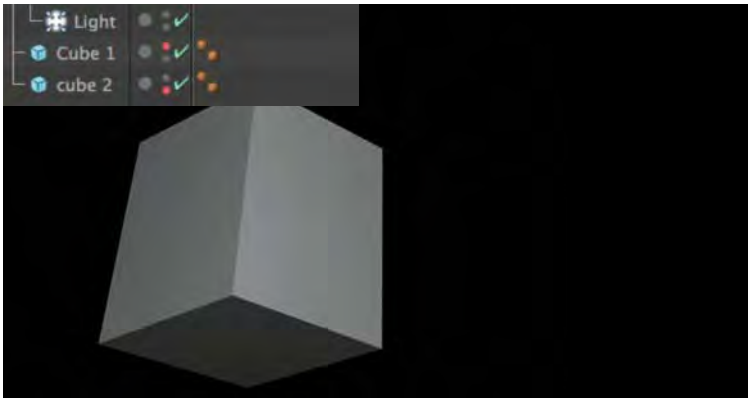
texture, together with shaders and other image effects – at this stage, of course, your image appears pretty simple (Illus. 48; you can find more on rendering in Chapter 16, Multi-pass Rendering and Compositing).

At present though, we are more interested in the very practical aspects of rendering, e.g. that you can hide objects from display in both Editor and rendering by clicking on the respective button in the Object Manager until it turns red. When the upper point appears red, the object is invisible in the Editor (Illus. 51), the lower point being red means the object will not be seen in the rendering (Illus. 52).
As you can see, an object such as Cube 2 may be visible in the Editor, but not be rendered – and vice versa, as is the case with Cube 1.

You may pool objects by grouping them – select both cubes in the Object Manager and select the Group Objects command from the Object menu (Illus. 53).
Now a so-called Null Object replaces the cube objects in the Object Manager. Click on the small cross beside its name in the list, and you can see the two objects again, having turned into sub-objects to the Null (Illus. 54). Their position is now related to the Null's position, as was the case with the light and camera before. You may hide the Null Object (which of course can be given a more meaningful name) from display too, the same way you dealt with its subordinate cubes before. When the Null itself is hidden, e.g. in the Editor view (Illus. 54), its subordinate objects may still be visible – the upper point next to the respective object (Cube 1 in your example), just has to be green.
The rules for the relationship between superior "red" and subordinate "green" objects applies to the rendering view accordingly.



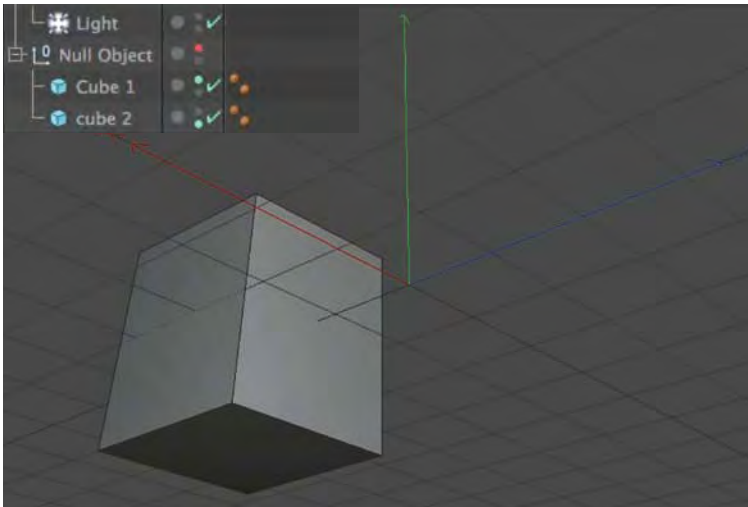
51

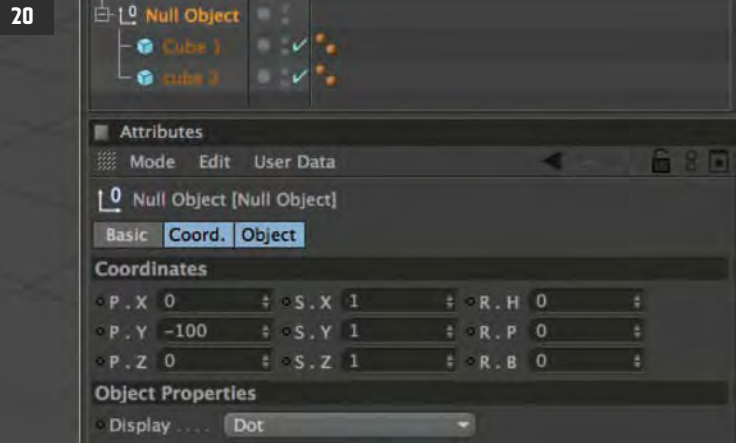


52

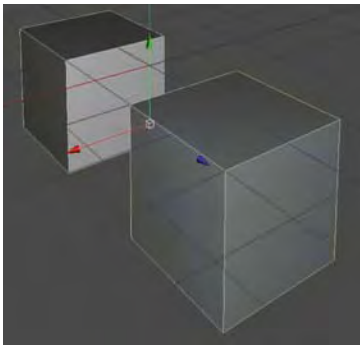
53

54

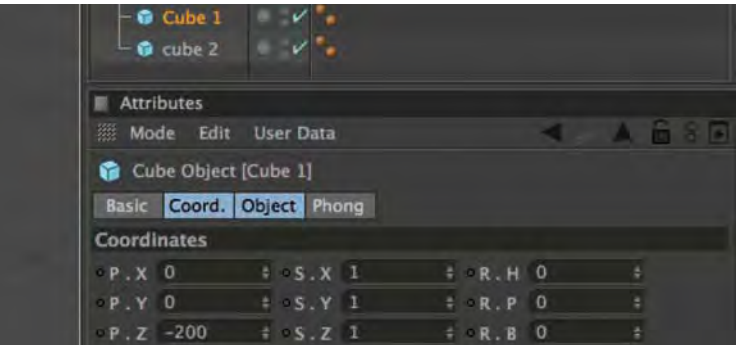




55



56



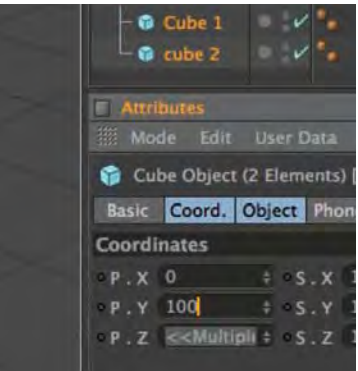
57

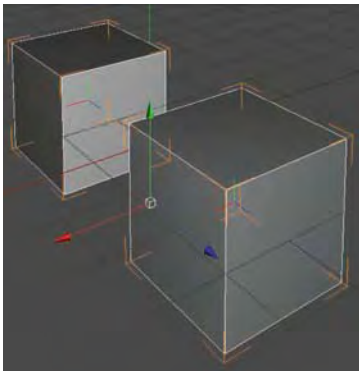
58

59

60

61





Let us continue with the relationship between superior and subordinate objects. Both the cubes and the superior Null Object have position coordinates (P.X, P.Y and P.Z in the Coordinates panel of the Attributes Manager).

If you change the position of the Null, e.g. by dragging it down (P.Y = -100, Illus. 55), the cubes will also slide to the new position (Illus. 56). Since they are subordinate to the Null, their position now refers to the zero point of the superior object, instead of still being related to the global zero point of the global scene.

You can verify this by clicking on one of the cubes and read its coordinates – although the Null has been moved together with its cubes, the Attributes Manager still shows P.Y = 0 for the sub-objects, (Illus. 57) – this is correct, since the cube and the Null had the same Y value before and nothing has changed in their relation after changing their position.

If you want to find out the global location of the cube, you have to take a look at the Coordinates Manager, not the Attributes Manager.

The Y position is initially shown with value 0 as well (Illus. 58), but if you select the World Option from the pop-down menu below the position values, you can read the global elevation of your cube (Illus. 59).

You may also edit values for a number of