CTR Stereoscopic Tips

2014/10/02

Version 2.0

The content of this document is highly confidential and should be handled accordingly.

Table of Contents

Figures

1 Introduction

This document was created (1) to describe methods for creating dramatic stereoscopic effects that are comfortable for users and (2) to point out issues that developers and content creators should be aware of. The document was written for creators of 3D content designed for display on a CTR system (such as movies and games that support stereoscopic effects) and for developers of CTR applications.

Chapter 2 Tips for Stereoscopic Display presents general information about images displayed on the CTR, such as characteristics of the CTR screen and human vision.

Chapter 3 Tips for 3D Content Creators provides information for creators of 3D content.

Chapter 4 Tips when Developing Games provides information for application developers.

2 Tips for Stereoscopic Display

This chapter describes useful techniques and important precautionary information common to both 3D content creation and application development.

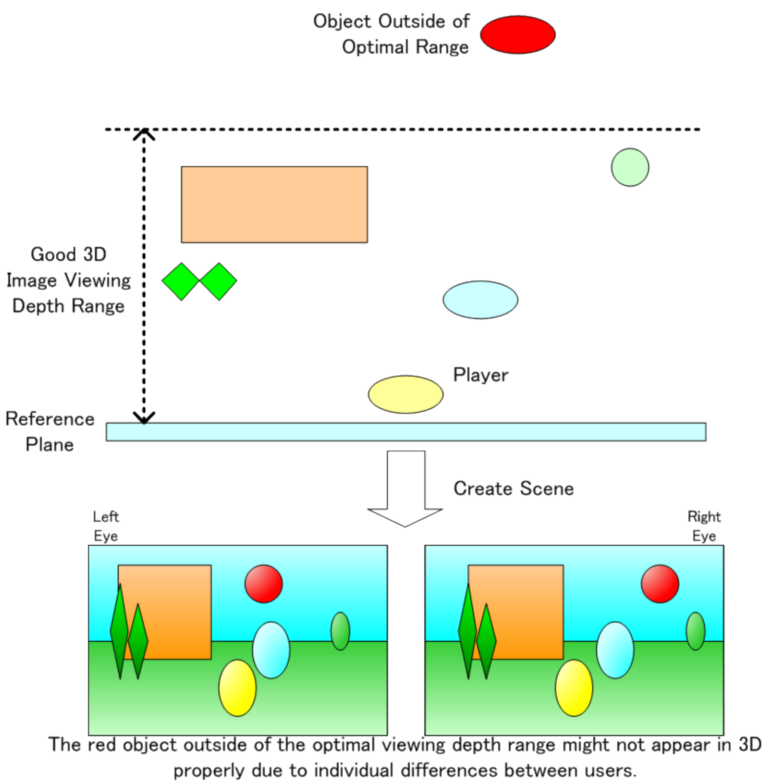
2.1 Optimal 3D Image Viewing Depth Range

A depth range of up to around 4 cm inwards from the surface of the LCD screen (in terms of parallax disparity, 7.5 mm or 39 pixels) is ideal for stereoscopic images on the CTR.

If you have several objects that you want to display simultaneously as 3D content, adjust the parallax disparity for all of the objects to place them within this optimal depth range and to position them in front of or behind each other as appropriate.

For instance, place the character controlled by the user slightly behind the reference plane (the surface of the screen) to make effective use of the viewing depth range that is comfortable for the user.

Figure 2-1 Optimal 3D Image Viewing Depth Range

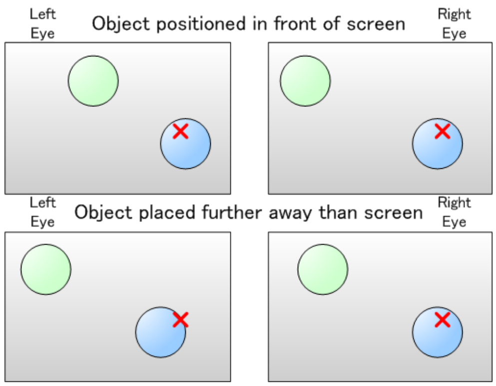


2.1.1 Positioning Displayed Objects

Placing objects in front of the reference plane (the screen surface) can cause visual inconsistencies, such as when dust, fingerprints, or reflections on the screen are seen in double vision through an object placed in front of the screen (items behind an object in stereoscopic view appear twice, once for each eye).

To combat this, place the object behind the reference plane, so that any surface dirt or dust is much more easily ignored such as when looking through a window or mirror.

Figure 2-2 Example of Surface Dirt Causing Discomfort to the User



Note: The red "x" marks in the figure indicate the location of a smudge or other dirt on the screen.

Objects placed in front of the reference plane and close to the upper or lower edges of the frame can appear as if they are abruptly cut off by the frame, which is further away relative to the object. Placing objects near the left or right edges of the frame can cause very strong visual discomfort due to the conflicting placement of the screen and the frame.

Consequently, Nintendo recommends designing your game so that objects cannot be displayed in front of the reference plane where they may stick out of the frame of the screen, or using some other means of ensuring that objects are never displayed sticking out of the frame when placed in front of the reference plane.

2.2 Optimal Camera Work for Stereoscopic Display

Objects displayed in front of the reference plane or beyond the depth range shown in section 2.1 Optimal 3D Image Viewing Depth Range may be difficult to see in 3D, or may not appear to be 3D at all, depending on individual user differences.

Suddenly placing a displayed object in front of the reference plane can make it difficult for the user to perceive the object stereoscopically. To make it easier for the user to perceive the object stereoscopically, move the object gradually from the reference plane towards the user.

Similarly, when placing a displayed object far behind the reference plane, first place the object within the optimal viewing depth range to allow the user to focus on it, and then move to the location further away from the plane.

Stereoscopic display thus requires a new approach to camera work, requiring that developers and content creators consistently keep in mind the position of the reference plane and the depth of objects relative to this, in addition to the elements that must be taken into account for conventional camera work.

Game developers in particular must consider the movement of the player's point of focus through all three dimensions, and continuously imagine where to place the frame of the LCD screen within the virtual space. For example, it might be convenient to build in a debugging process wherein developers themselves can see the reference plane in the virtual space and the depth information in the real world.

2.3 Suppressing Crosstalk

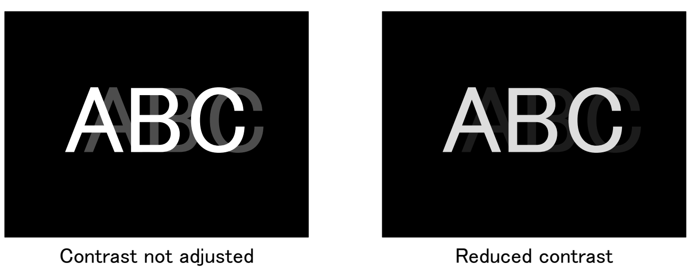
Crosstalk is the phenomenon where an image for one eye becomes visible to the other eye, resulting in double vision for the user. This can occur in particular when displaying a light object against a dark background, such as white text on a black background.

The parallax barrier method used in the CTR system relies on optical properties such as diffraction, scatter, and barrier transparency, and cannot completely suppress crosstalk. However, it is possible to minimize crosstalk by controlling the contrast between the background and foreground objects.

Crosstalk only occurs when displaying content that has a parallax disparity. When you want to avoid crosstalk for certain content (such as the player-controlled character), you can minimize both this parallax disparity and crosstalk by displaying the content on the reference plane. However, this approach can lead to portions of the object appearing in front of the reference plane, thereby running into the issues described in sections 2.1 Optimal 3D Image Viewing Depth Range and 2.1.1 Positioning Displayed Objects.

Choose the approach you wish to emphasize based on your game content.

Figure 2-3 Example of Crosstalk and Suppressing Crosstalk



3 Tips for 3D Content Creators

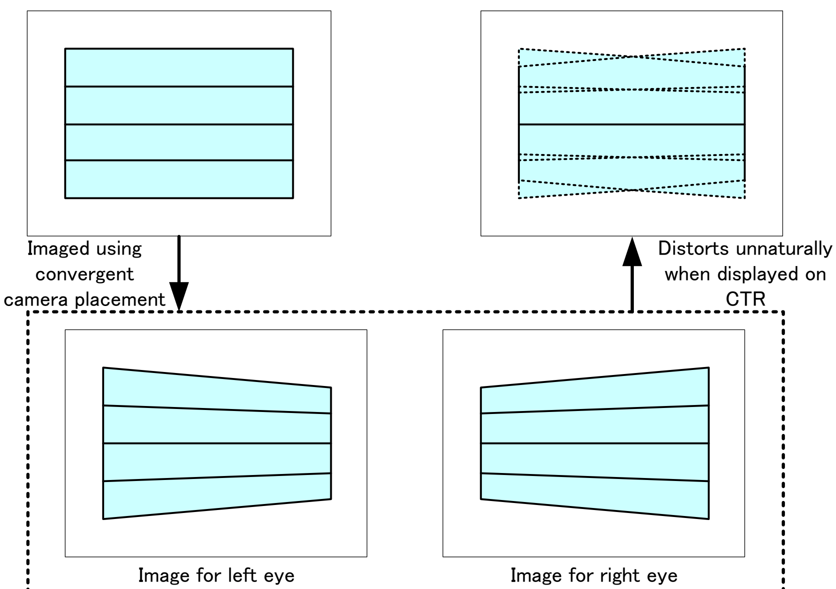
This chapter describes useful techniques when creating 3D content and important precautions to be aware of.

3.1 Using Parallel Cameras

Stereoscopic display on the CTR displays separate images for the left and right eyes on a single flat surface to generate the illusion of three dimensions. Consequently, positioning two cameras at an inward angle instead of parallel to each other to display 3D images created with convergent lines of sight will result in unnatural distortion in the final composite image. Nintendo therefore recommends using parallel camera placements when rendering 3D computer-generated video. For live-action 3D movies shot using convergent stereoscopic camera, make sure that this distortion is within acceptable limits.

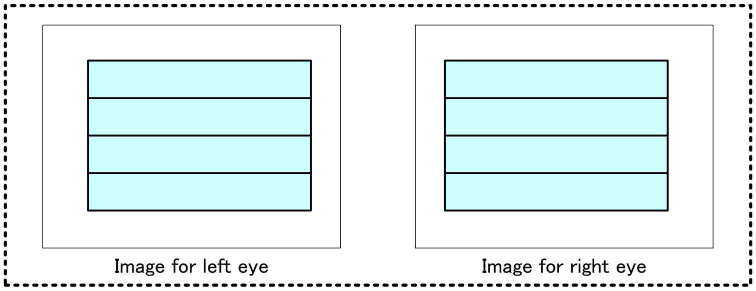
The figure below shows an example of the distortion that occurs when the stereoscopic cameras use a convergent line-of-sight placement to view a plane. This example uses an extreme angle of convergence for purposes of illustration.

Figure 3-1 Unnatural Distortion when Using Convergent Camera Placement



The figure below shows the same scene, using a parallel camera placement.

Figure 3-2 Imaging Using Parallel Camera Placement

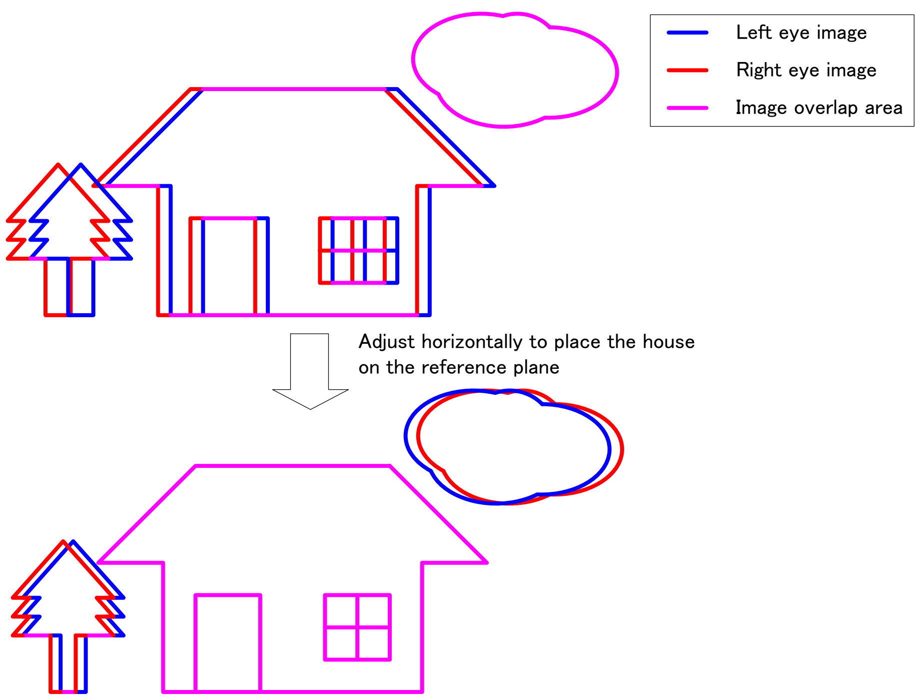


3.2 Adjusting the Parallax Disparity of Parallel Images

Compared to convergent camera placement where the resulting image has objects appearing both in front of and behind the reference plane, the reference plane for images taken using parallel camera placement is always the background, and everything else appears in the foreground. However, it is possible to shift the left and right images horizontally to adjust the position of the reference plane so that objects appear both in front of and behind the reference plane.

The image below shows an example of adjusting the horizontal position of the left and right images taken using parallel camera placement.

Figure 3-3 Horizontally Adjusting Parallel Images



3.3 The Differences from 3D television and 3D Movies

The Nintendo 3DS display is viewed from a distance of around 30 cm from the eyes, for a very different physical setup and balance between viewing comfort and sense of depth compared with 3D television or 3D movies in a theater, where the distance could be several meters to several dozen meters.

When creating 3D content, Nintendo recommends that you double-check its appearance on a Nintendo 3DS test unit.

3.4 Left-Right Image Synchronization

If the left and right images in a 3D video become even slightly desynchronized, completely different left and right images might be displayed (such as when the game scene changes). This can result in strong user discomfort. Also, objects moving up and down appear vertically distorted or disconnected between the left and right images. Even slight vertical distortions between the left and right images can result in discomfort for the human eye.

If you use a consistent system for creating 3D videos (either taking 3D video or rendering scenes, saving them as 3D videos, and then encoding them) and this system can guarantee left-right image synchronization, no further work is required. However, if you are manually creating or editing 3D videos, “frame desync” (when the left and right images are not synchronized) can sometimes occur. Nintendo recommends establishing a system for checking that the final version of 3D content is free of frame desync.

Comparing static versions of the left and right images side by side for each and every frame is the recommended way to check for frame desync. This is because the degree to which frame desync causes discomfort when the content is viewed in full 3D can vary between individuals and may result in it being overlooked.

4 Tips when Developing Games

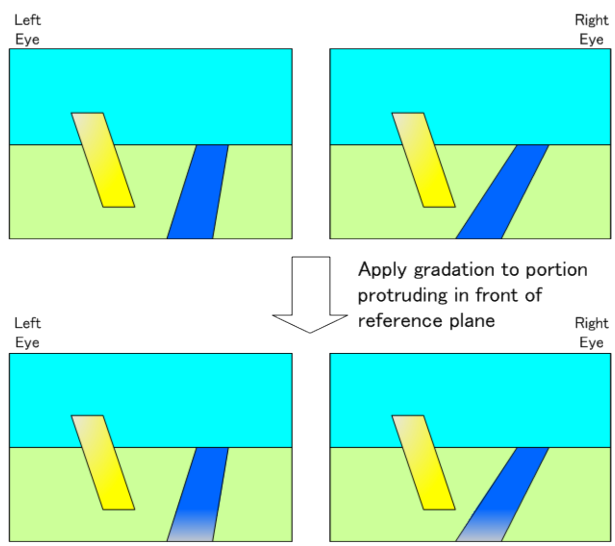
This chapter describes useful techniques for developing games using stereoscopic display. It also describes precautions that need to be kept in mind.

4.1 Displaying Objects in Front of the Reference Plane

As noted in section 2.1.1 Positioning Displayed Objects, objects located in front of the reference plane that are also close to the frame of the screen can cause very strong feelings of visual discomfort in viewers.

If displaying anything protruding towards the user such that it might overlap with the frame of the screen, Nintendo recommends blurring the image where it touches the frame in order to minimize cases where parallax disparity becomes too great. When applying a gradation to image portions near the frame, use gray instead of black to achieve convergence over short widths.

Figure 4-1 Example of Gradation



Even if the displayed objects do not touch the frame of the screen, it is difficult to make objects located in front of the reference plane appear in 3D because the parallax disparity is large, and such images are likely to cause discomfort in users.

However, when characters and the camera move freely in conjunction with commands from the user, there are cases in which, depending on the game design, displaying objects in the region in front of the screen is unavoidable. In such cases, you can use the near clipping feature to display objects where they do not protrude any further out of the screen than a fixed position.

When near clipping cannot be used because, for example, it might result in an undesirable cross-sectional appearance for an object, you can alter the transparency of the object depending on distance, so that the transparency is higher the further the object is in front of the reference plane. In this way, you can reduce the characteristic discomfort for users associated with stereoscopic display of objects displayed in front of the reference plane.

This method of altering transparency is used in the *Dillon's Rolling Western* Nintendo 3DS downloadable software.

4.2 Displaying Distant Objects

Objects displayed very far away from the reference plane may be difficult for users to perceive as 3D due in part to the users' eyes focusing on the screen's surface, which is far removed from the optimal viewing depth range, and in part to the difficulty in displaying the left-right parallax disparity required to generate stereoscopic perception for distant objects.

There are two main reasons for this difficulty in generating left-right parallax disparity as objects become more distant:

- The limits of the LCD screen's resolution make it impossible to display increasingly distant objects with the fine detail required.
- The angles of view for multiple objects displayed in the left and right cameras become effectively the same as the objects become more distant from the cameras.

The outlines of small displayed objects may be unstable and appear to flicker. Alternately, big objects that have the same angle of view in both cameras will appear to lose surface detail, even though the user still perceives them as distant.

Note that widening the angle of view between the two cameras by more than is necessary can cause distant objects to appear smaller. A giant enemy or other object displayed at a distance may also appear to have smooth surfaces or even completely flat surfaces in some cases, even though the object as a whole is clearly visible in the distance.

As noted above, there are limits to the stereoscopic display generated through left-right parallax disparity. If you run into these limits, you can use a combination of parallax disparity and the following left-right image differences that do not depend on parallax to heighten the sense of 3D realism.

- Add (exaggerate) shadows
- Use an object layout within the virtual space that clarifies perspective
- Use aerial perspective
- Blur objects that are closer or further away than the objects of focus in the scene (similar to photographic depth of field)
- Use motion parallax (difference in movement speed)
- Overlap objects to clarify depth

4.3 When Using the ULCD Library

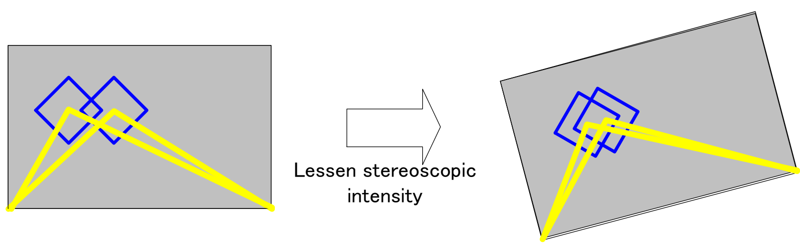
Render using the camera matrix generation functions provided by the CTR-SDK's ULCD library to display objects using a parallel camera placement. When using these functions, Nintendo recommends (1) setting the reference plane so that the player's character does not appear in front of the LCD screen, (2) and modifying the argument (`Dr` adjustment coefficient) to adjust depth intensity for each scene to maintain this reference plane positioning.

4.4 Adjusting Stereoscopic Intensity by Detecting Console Tilt

When using 3D display for action games or other games that might involve energetic movement is to use the gyro sensor to detect console tilt or rotation and adjust stereoscopic intensity accordingly.

Face Raiders, the built-in mini-game, uses this approach.

Figure 4-2 Adjusting Stereoscopic Intensity by Detecting Tilt



4.5 Adjusting the Stereoscopic Effect

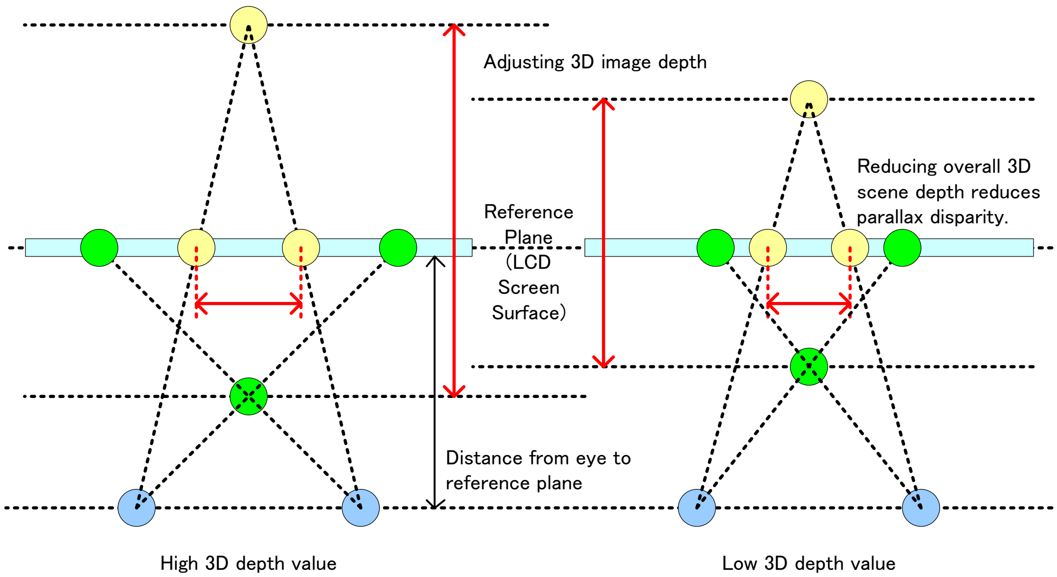
In order to make a 3D game that most people can fully enjoy, you must implement some means of adjusting the intensity of the stereoscopic effect in order to account for differences between individual users.

The Nintendo 3DS includes the 3D depth slider by which users can adjust the intensity of the stereoscopic effect on the hardware level. Support this in your application to provide a simple way for users to adjust this effect to their own preferences for all games. Use the ULCD library provided in the CTR-SDK to automatically support the 3D depth slider without having to implement any special handling code.

There are two ways to adjust the stereoscopic effect. One way, using the 3D depth slider, is to change the depth of the scene from front to back by increasing or decreasing the front-back distance between scene objects and the reference plane, without changing the relative positioning of the reference plane and objects along the depth axis. The other way is to change only the position of the reference plane without changing the relative positions and distances of the displayed objects along the depth axis.

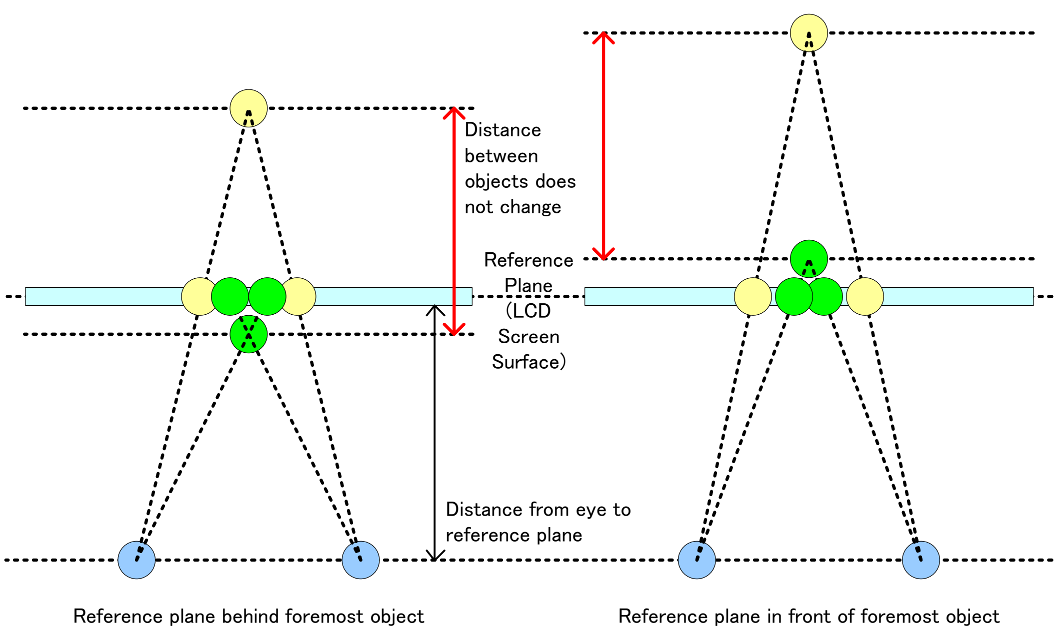
The following figure illustrates how the 3D depth slider adjusts the scene depth.

Figure 4-3 Adjusting Scene Depth Using the 3D Depth Slider



The following figure illustrates how moving the reference plane changes the stereoscopic effect. This moves an object from in front of the reference plane to behind it, but does not change the relative distance between the two objects along the depth axis.

Figure 4-4 Adjusting Reference Plane Position



4.6 Preventing Frame Desync

As described in Section 3.4 Left-Right Image Synchronization, frame desync (when the left and right images for a game frame are not synchronized) typically causes strong user discomfort. This section describes items to keep in mind to prevent frame desync when designing 3D game content.

4.6.1 Only Display Frames After Both the Left and Right Images Are Rendered

If you display the left image as soon as it is rendered and then move on to rendering the right image, the images might not always be updated in the same game frame if the frame rate drops for some reason.

Similarly, updating the right image in even-numbered frames and updating the left image in odd-numbered frames to try to reduce computational load is not appropriate because the Nintendo 3DS displays content at 60 FPS. Even the offset of 1/60th of a second between the left- and right-eye images in 3D content created using this method is enough to cause discomfort in users.

4.6.2 Use Static Screenshots to Check for Left-Right Image Synchronization

Nintendo recommends taking static screenshots of the left-eye and right-eye images displayed on the upper screen in each game frame and then comparing them frame by frame. Adding this step to your debugging workflow can help catch unexpected frame desync caused by bugs in the game.

Frame desync during scene changes is usually relatively obvious, but frame desync that occurs within a single scene is more subtle and sometimes hard to detect. One easy way to catch this type of frame desync is to apply the screenshot method described above to sequences in which a game object is moving up and down and then checking whether the object has exactly the same height (in y coordinates) in both the left and right images.

Finally, Nintendo recommends using the screenshot method to check for frame desync in 3D movies and other in-game cutscenes during which the application is running other processes.

Revision History

Version	Revision Date	Category	Description
2.0	2014/10/02	Additions	• 3.4 Left-Right Image Synchronization • 4.6 Preventing Frame Desync
1.1	2012/02/28	Changes	• 3.2 Adjusting the Parallax Disparity of Parallel Images Modified color of lines in figure. • 4.1 Displaying Objects in Front of the Reference Plane Added method of increasing the transparency of objects in front of the reference plane. • 4.4 Adjusting Stereoscopic Intensity by Detecting Console Tilt Removed redundant description.
1.0	2011/11/08	—	Initial version.