3D Interface Specifications

White Paper

Philips 3D Solutions

Document Information

Info	Content
Title	3D Interface Specifications, White Paper
Date	21 December 2006
Security	The attached material and the information contained herein are proprietary to Philips 3D Solutions. Copying, reproduction, adaptation, modification or dissemination in whole or part is not permitted without written permission from Philips 3D Solutions.
Contact address	Philips 3D Solutions High Tech Campus 27 5656 AE Eindhoven The Netherlands, E-mail: 3DSolutions@philips.com Website: www.philips.com/3dsolutions

Table of Contents

1 Introduction 5
1.1 Scope and Purpose 5
1.2 References 5
1.3 Notations 6
2 Physical interface 7
3 Display Control Tool 8
4 3D Frame 9
5 Video data 11
5.1 2D and Z sub-images 11
5.2 2D sub-image 11
5.3 Z image 12
6 Header 13
6.1 Header location in the 3D frame 13
6.2 Header data 15
6.3 Content 17
7 Operation modes 18
7.1 2D Mode 18
7.2 3D Mode 18
8 Stretching of images 20
9 Converting depth to disparity 22

1 Introduction

1.1 Scope and Purpose

This document contains the interface specifications of the 3D Displays as developed by Philips 3D Solutions. This document covers the physical specification as well as the data format.

All displays with type numbers 42-3D### and 20-3D### developed by 3D Solutions comply with the specification described herein.
The header described in section 6 is only supported by displays with firmware 10.7 or higher. Please check your firmware version using the Display Control Tool provided with each Philips 3D display.

The intended audiences are:
3D system integrators that do not make use of software from Philips 3D solutions. The intended applications are 3D CAD design, gaming, gambling, interactive applications, etc. For these persons the sections 6.1general header location and 6.2 header description are most interesting.
For the development of digital signage related software we refer to section 6.1 where the general header location is denoted and to section 6.3 where the standard headers are mentioned for the 3d content using the s3d, v3d or b3d extensions.
For the development of other software, please first contact Philips 3D Solutions.

Philips is under no circumstances responsible for malfunctioning of third party software. Furthermore, Philips is not liable for the consequences resulting from changes of mistakes within this whitepaper.

1.2 References

The following references are only informative.

[DCT]	Display Control Tool; User manual; Philips 3D Solutions;
[DDC]	VESA Display data channel standard; Version 3; December 15, 1997;
[DDC/CI]	VESA Display data channel command interface (DDC/CI) standard; Version 1; august 14, 1998;
[DPMS]	VESA Display Power Management Signaling (DPMS) standrad; version 1.0; revision 1.0; Augustus 20, 1993;
[DVI]	Digital Visual Interface DVI; Digital Display Working Group; Revision 1.0; 02 april 1999;
[EDID]	VESA enhanced extended display identification data standard; Release A, Revision 1; February 9, 2000;
[GTF]	VESA Generalized timing formula (GTF) standard; version 1.0; revision 1.0; December 18, 1996;
[I2C]	The I2C bus specification; Version 2.1; January 2000; Philips Semiconductors;
[3DTVC]	"3D throughout the video chain" by B. Barenbrug, proceedings of ICIS 2006 (pp 366-369).

1.3 Notations

0xNN	Hexadecimal numbers are represented by using 'C' language notation.
0bNN	Binary numbers are represented by using 'C' language notation.
NN	Decimal numbers have no prefix.

2 Physical interface

The connection between the host and the 3D Display makes use of a DVI based interface, see [DVI].

Table 1. DVI specifications for the 42” displays

Parameter	Value
Resolution	1920 x 1080
Refresh rate	60 fps
Gamma	No gamma correction.
Reference white	Wx: 0.285 Wy: 0.293 (at 9300°K)
Aspect ratio	16:9
Order of sample scanning	Left to right. Top to bottom. Progressive
Coded signal colour space	RGB (8 bit coding)
Supported standards	[DDC], [DDC/CI]
Not supported standards	[GTF], [DPMS]

Table 2. DVI specifications for the 20” displays

Parameter	Value
Resolution	1600 x 1200
Refresh rate	60 fps
Gamma	No gamma correction.
Reference white	Wx: 0.313 Wy: 0.329 (at 6500°K)
Aspect ratio	4:3
Order of sample scanning	Left to right. Top to bottom. Progressive
Coded signal colour space	RGB (8 bit coding)
Supported standards	[DDC], [DDC/CI]
Not supported standards	[GTF], [DPMS]

3 Display Control Tool

The Display Control Tool is supplied with each 3D Display from Philips 3D Solutions and is running on Windows based computers. It enables the viewer to control the perceived depth and color settings real-time. It is advised to always install the Display Control Tool on each computer.

The Display Control Tool communicates with the display via the DVI cable making use of the [DDC/CI] protocol. There is no interaction on the computer between the Display Control Tool and any other software. So while developing software according to these interface specifications no consideration regarding the Display Control Tool must be taken.

The Display Control Tool requires an NVIDIA graphics card.

See [DCT] for more information about the Display Control Tool.

4 3D Frame

The resolution of a frame that is sent to the display via the DVI cable has a resolution of 1920x1080 for 42" or 1600x1200 for 20". A frame contains the following data:

- 2D sub-image with a resolution 960x540 for 42" or 800x600 for 20"
- Z sub-image with a resolution 960x540 for 42" or 800x600 for 20"
- Header

From now on we continue only with the explanation of 42" (1920x540=2 images of 960x540) The reader interested in 20" display has to read 1600x600= 2 images of 800x600. The rest is stays the same.

This is a very flexible format. Among the advantages of using 2D and Z images (compared to for example 9 separate views) are:

- user control over the amount of perceived depth (controlled via for example the header or via the Display Control Tool)
- (future) compatibility with displays which have a different number of views than the current 9, or have a different lens design which requires different interleaving (see Section 7.2), allowing the display to render in a way optimized for its optical components.
- suitability for compression for content storage and distribution
- advantages in content editing, because the disparity is explicit

More information can be found in [3DTVC]. Note: that whenever 2D and Z is mentioned the Z map describes the disparity and not the depth. The picture below shows how the header and the 2D and Z sub-images are organized in a frame of 1920x1080.

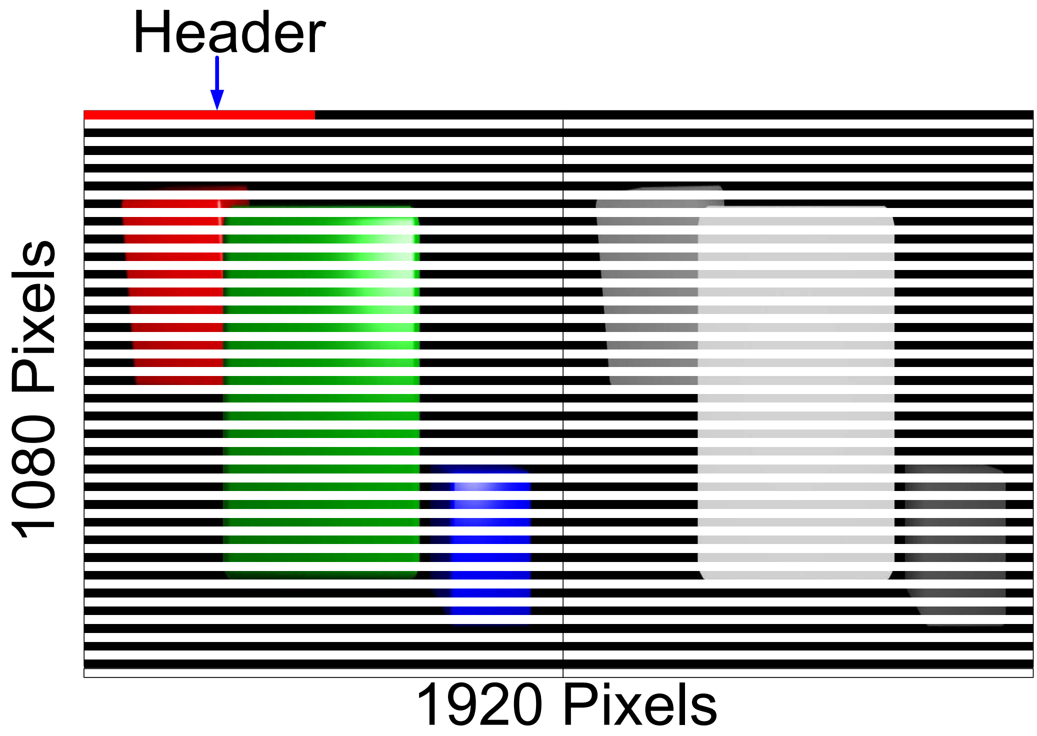

Figure 1. Overview of 3D frame.

The 2D and Z sub-images of 960x540 are positioned besides each other, which result in a total line width of 1920 pixels. Between each video line a blank line is added. This doubles the vertical resolution from 540 to 1080. Chapter 5 explains the 2D and Z sub-images in more detail.

A header is located in the upper left corner. The function of the header is twofold. When the display detects the header it switches to 3D mode and the header contains settings for rendering processing. The header is further explained in chapter 6.

5 Video data

5.1 2D and Z sub-images

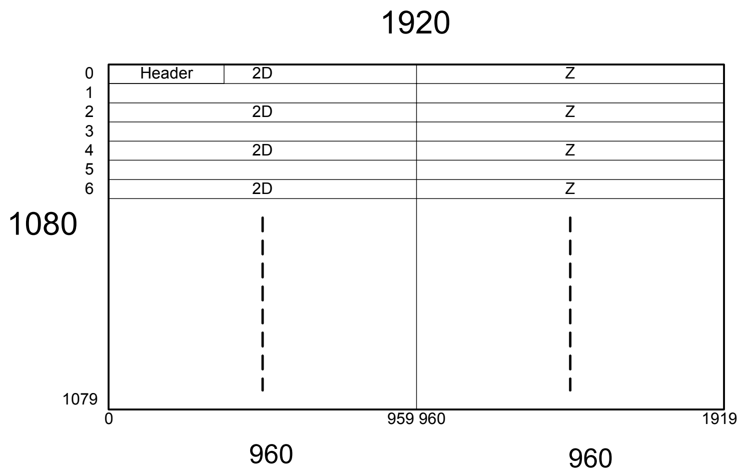

Figure 2. 3D frame layout.

The picture above shows the layout of a 3D frame. It contains a 2D sub-image and a Z sub-image both with a resolution of 960x540[1]. The 2D sub-image is positioned in the left half of the frame, and Z sub-image in the right. A blank line is inserted below each line of the 2D and Z.

5.2 2D sub-image

Properties of 2D sub-image:

- The 2D sub-image has a resolution of 960x540.
- The 2D image is an R,G and B image with 8 bits per sub-pixel.
- No gamma correction is performed in the display.

[1] Like in chapter 4, the resolutions are for the 42" display. You should read 1600x1200 instead of 1920x1080 and 800x600 instead of 960x540 for the 20" display.

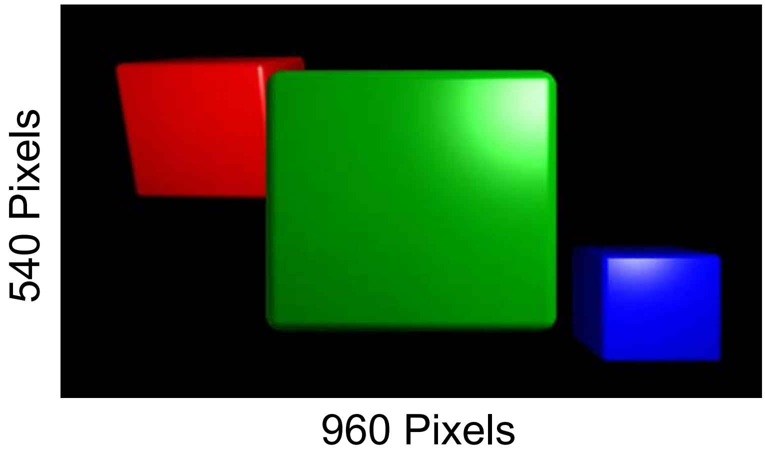

Figure 3. 2D sub-image example

5.3 Z image

A Z image contains disparity values with a range of 0 to 255, where a value of 0 corresponds with objects located with a maximum disparity behind the screen and 255 corresponds with objects located closest to the observer.

An implementation is that the Z image is a black and white image. This means that the R, G and B sub-pixels have the same value per pixel. However the display only uses the red sub-pixels. The green and blue sub-pixels are discarded.

Sometimes a Z image is also called 'depth map' but note that it describes the **disparity**.

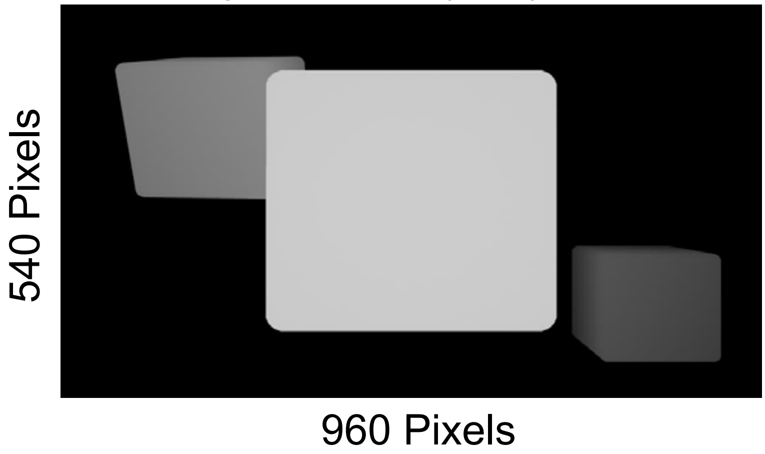

Figure 4. Z sub-image example

6 Header

The header described in this section is only supported by displays with firmware 10.3 or higher. Please check your firmware version using the Display Control Tool provided with each Philips 3D display.
The header is positioned in the upper left corner of a frame. Each frame will contain a header otherwise the display will interpret it as a 2D frame. The header instructs the display which viewing experience is required.

6.1 Header location in the 3D frame

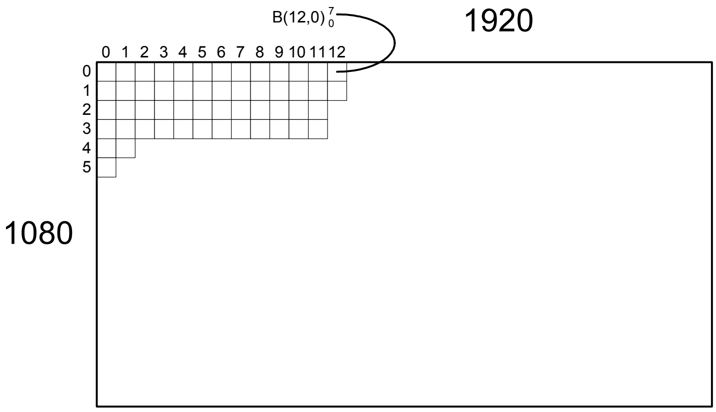

Figure 5. Numbering of rows and columns of screen.

The figure above shows how the columns and rows are numbered. Numbering starts with zero in the upper left corner. Each square pixel consists out of an R, G and B sub-pixel.
$B(12,0)_0^7$ Denotes the blue byte in the upper row in the 13th column and bits 7 down to 0. With 7 the Most Significant Bit.

The header is located in the blue sub-pixels. It is required to give all the 8 bits in the blue sub-pixel the value of the header bit. In this way the system becomes more tolerant to changes in brightness, contrast settings and noise. The header starts at the first pixel in the first row ($B(0,0)$) and onwards ($B(x,0)$). Only in the even blue sub-pixels the header is located, starting with 0, 2, 4 etc.

During rendering, the blue sub-pixels that contain the header are replaced by the blue component of their neighboring pixels. This masks the header entirely, while it hardly affects image quality.

The following formula shows which bit of which sub-pixel byte forms which bit (y) and byte (x) of the header ($x \in [0,9]$, $y \in [0,7]$):

$$B(2 \cdot (7-y) + 16 \cdot x, 0)^7 = H(x)^y \quad [1]$$

21 December 2006

The following translation table shows how the header is composed from bits in the blue sub-pixels. The first header byte, $H(0)$, is composed by combining blue sub-pixel bits.

Table 3. Translation table of blue sub-pixel MSB bits into header bytes.

Sub-pixel bit	Header bit
$B(0,0)^7$	$H(0)^7$
$B(2,0)^7$	$H(0)^6$
$B(4,0)^7$	$H(0)^5$
$B(6,0)^7$	$H(0)^4$
$B(8,0)^7$	$H(0)^3$
$B(10,0)^7$	$H(0)^2$
$B(12,0)^7$	$H(0)^1$
$B(14,0)^7$	$H(0)^0$
$B(16,0)^7$	$H(1)^7$
$B(18,0)^7$	$H(1)^6$
$B(20,0)^7$	$H(1)^5$
$B(22,0)^7$	$H(1)^4$
$B(24,0)^7$	$H(1)^3$
$B(26,0)^7$	$H(1)^2$
$B(28,0)^7$	$H(1)^1$
$B(30,0)^7$	$H(1)^0$
$B(32,0)^7$	$H(2)^7$
$B(34,0)^7$	$H(2)^6$
$B(36,0)^7$	$H(2)^5$
etc.	etc.

6.2 Header data

The header is 10 bytes long. The table below lists the content of these 10 bytes, which is further defined in sections 6.2.1to 6.2.6.

Byte	Content
$H(0)$	Header_ID
$H(1)$	Hdr_Content_type
$H(2)$	Hdr_Factor
$H(3)$	Hdr_Offset_CC
$H(4)_6^7$	Hdr_Select
$H(4)_0^5, H(5)$	Reserved
$H(6) - H(9)$	EDC

Unused and reserved bits **must** be set to zero.

The display interprets the header each frame, 60 times per second. Changed header values are effectuated directly.

6.2.1 Header_ID

Indicates the format of the remainder of the header see table 4.

Table 4. Header_IDs

Byte	Content	Description	Value
$H(0)$	Header_ID	This header ID	11110001

6.2.2 Hdr_Content_type

This value defines the kind of content. Based on this the various visualization parameters, like factor and offset, are chosen that influence the rendering process.

Table 5. Content type

Byte	Content	Type value (7:0)	Type of Content
$H(1)$	Hdr_Content_type	00000000	No depth
		00000001	Signage
		00000010	Movie
		00000011	Game
		00000100	CGI
		00000101	Still
		00000110	Reserved

		etc.	

6.2.3 Hdr_Factor

Table 6. Header factor value

Byte	Content	Description	Default	Range
$H(2)$	Factor	Percentage of the display recommended disparity value. (Factor/64)	64	[0-255]

Each 3D Display has a 'Display recommended depth value', which corresponds to an acceptable maximum depth factor value for that specific type of display. This value strongly depends on the lens design. The factor field in the header contains the percentage to be used from the display recommended depth value. The value of 64 corresponds with the 100% of the display recommended depth value. It is allowed to use values higher than 64. The factor works on a linear scale and is multiplied with the factor controlled by the user in the Display Control Tool.

6.2.4 Hdr_Offset_CC

Table 7. Header offset value

Byte	Content	Description	Default	Range
$H(3)$	Offset	Amount of range behind the screen. 0: Range is shifted in the direction of the viewer. 128: Range is equally divided in front and behind the screen. 255: Range is shifted away from the viewer.	128	[0,255]

Values in the Z map equal to the header-offset value will be located on the plane of the display. All values in the disparity map with a higher value will de displayed in front of the display.

Offset_CC is the offset controlled by the Content Creator. In the system there is also an Offset_user present, which is controlled by the user using the Display Control Tool.

6.2.5 Hdr_Select

Table 8. Header select

Byte	Content	Description
$H(4)^7$	Hdr_Factor_select	'1': Hdr_Factor is used; '0': Hdr_content_type is used;
$H(4)^6$	Hdr_Offset_CC_select	'1': Hdr_Offset_CC is used; '0': Hdr_content_type is used;

When all select signals are low the rendering settings are set to optimal settings for the content type denoted by Hdr_content_type. By making select signals high the settings for Factor and Offset_cc can be controlled individually by the header.

6.2.6 Reserved

These bits are reserved for future use and should be set to 0 to maintain compatibility with future systems.

6.2.7 EDC

The 4-byte EDC field contains an Error Detection Code computed over the first 6 header bytes. This EDC is also known as CRC-32 as defined in IEEE 802.3 and ITU-T V.42. Considering the Data Frame as a single bit field, starting with the most significant bit of the first header byte ($H(0)^7$) and ending on the least significant bit of the last header byte ($H(9)^0$), then the MSB bit is b79 and the LSB bit is b0.

Each bit b_i of the EDC is shown as follows for i = 0 to 31:

$$EDC(x) = \sum_{i=0}^{31} b_i x^i = I(x) \bmod G(x) \qquad [2]$$

Where: $$I(x) = \sum_{i=32}^{79} b_i x^i \qquad [3]$$

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1 \qquad [4]$$

6.3 Content

In many applications a standard header is sufficient. E.g. for our *.v3d, *.s3d and *.b3d files the 3dsmediaplayer uses a standard header (as denoted in table 9), that can be very easily applied.

Table 9. Standard headers

Extension	Header	Description
s3d	F10140800000C42DD3AF	Used for signage content.
v3d	F102408000001F3A7B38	Used for video content.
b3d	F10540800000E4D9502C	Used for stills.

We refer to section 6.1 for more information on the location of the header and how to code it into the image.

21 December 2006

7 Operation modes

The display has two operational modes, i.e. 2D mode and 3D mode, which are described in sections 7.1 and 7.2 respectively.

7.1 2D Mode

At the moment that the display detects no header it switches back to 2D mode. The 2D mode is achieved by placing the 1920x1080 pixels on the display after a image processing step that lowers the visibility of the optical coupling between the LCD and the lens layer. Thereby a high quality 2D image is visualized.

7.2 3D Mode

The image processing within the display takes place in 3 steps, see Figure 6.
First a frame with a header and a 2D and Z sub-image is applied to the DVI input connector. A demultiplexing block decomposes the 3D frame into the header and the 2D and Z sub-images. These 3 components are applied to the rendering block.
The rendering block generates 9 images, which have all a slightly different camera positions. The amount of perceived depth and other depth related parameters are controlled by the values in the header.
The 9 different images are fed to the interweaving block. This block ensures that each sub-pixel is exactly located under the right lens, which ensures the best 3D experience. The interweaving process is optimized for the optical behavior of the lens layer.

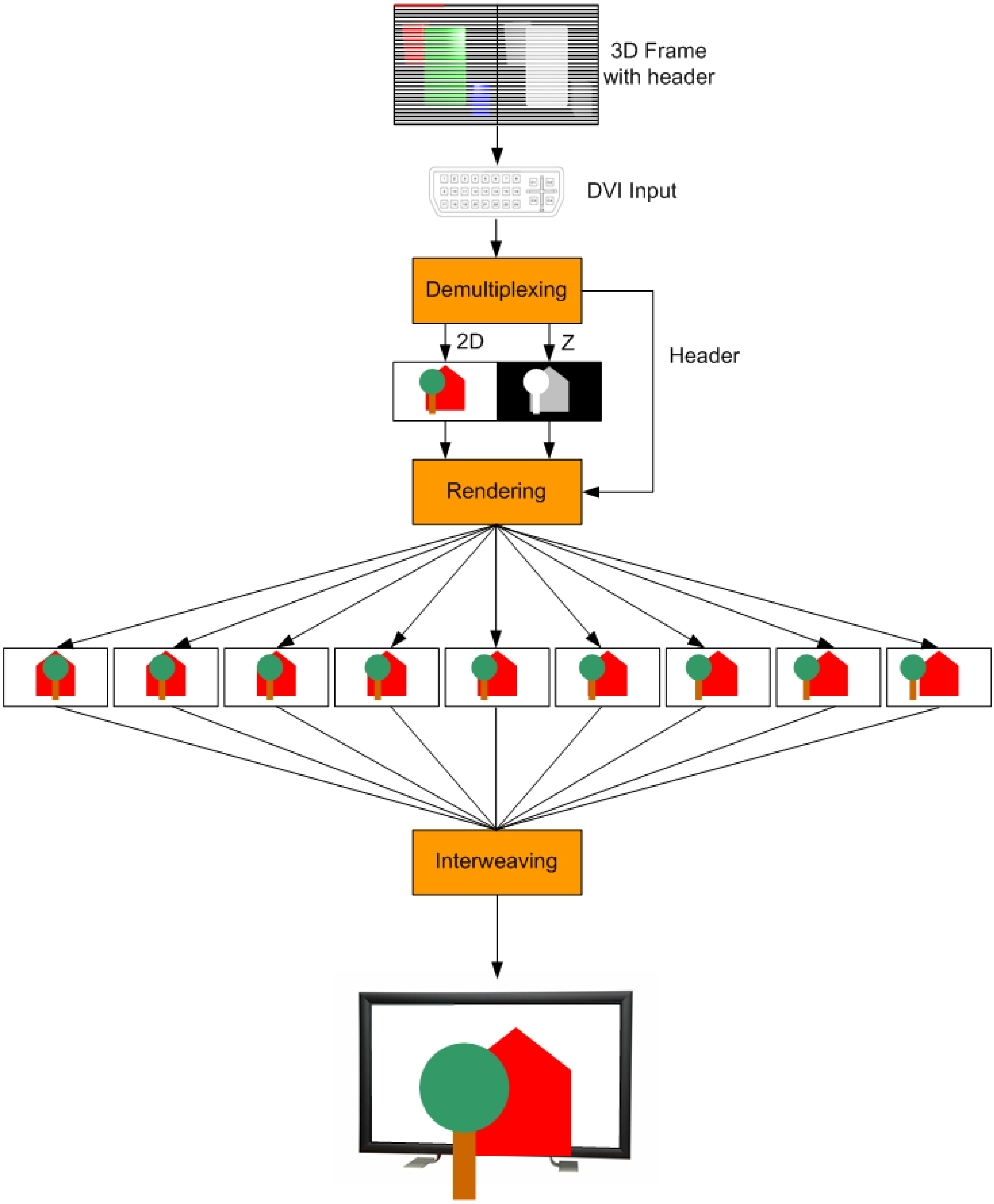

Figure 6. Overview of 3D image processing in the display.

8 Stretching of images

If an object positioned at the left edge of the image has a depth that puts it behind the screen, a viewer looking at it from the right would expect to see more pixels to the left of the object where the frame of the display would no longer be concealing the object. Because these pixels are not in the original image, the image is stretched slightly before rendering to prevent this problem. Figure 7 shows how the image is stretched. Stretching and rendering takes place inside the display.

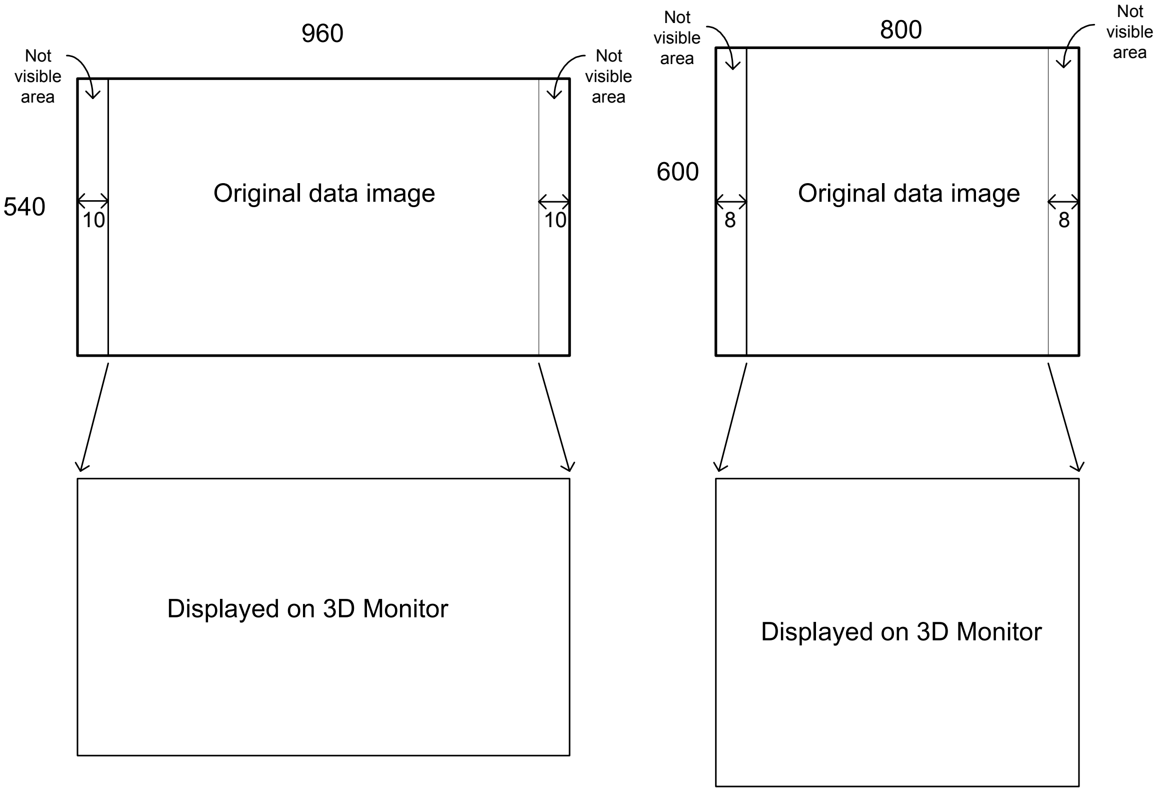

Figure 7. A band at the left and right side of the image data are not displayed on the 3D Display.

The original 2D input data has a resolution of 960x540. The leftmost pixels and the rightmost pixels of each row are not visible on the display, when the depth of the pixels is on the screen. However when the depth is behind the screen, the observer is able to see this information in these bands.
So while sending content to the display the designer must keep in mind that small bands at the left and right side will mostly not be visible. This holds especially for text and logos that are closely located to left and right borders of the screen.
A side effect of this stretching is that the aspect ratio of the content is somewhat distorted.

Table 10 Sizes of the "invisible" borders

Display	Pixels on the left	Pixels on the right
42" display, firmware version > 9.6	10	10
20" display, firmware version > 8.0	8	8

9 Converting depth to disparity

In 3D application a 3 dimensional (Cartesian) coordinate system is used to model objects. Normally these models are used to generate 2D images of scenes like: games or other applications. The function below describes the translation from depth (Z) to disparity (D(Z)). Herein depth describes the depth extracted from the application and needs to be normalized between 0 and 1. Further disparity refers to the difference in images from the left and right eye that the brain uses as a binocular cue to determine depth or distance of an object. The function below will describe the correct translation between depth and disparity:

$$D(Z) = M * \left(1 - \frac{vz}{Z - Zd + vz}\right) + C$$

where:

$D : disparity \quad [0,255]$

$Z : depth \quad [0,1]$

M : Linear function multiplier

Zd : depth of display plane

vz : View distance in coordinate units

C : Linear function constant

Within this formula there are a number of constants present M, Zd, vz and C. To obtain the best 3D performance for each of our displays use the correct values from the table below:

Constant	42"	20"
Zd	0.467481	0.459813
Vz	7.655192	6.180772
M	-1960.37	-1586.34
C	127.5	127.5

There are two kinds of 42" displays WOW and Comfort; both displays use the same constants within the conversion function.

