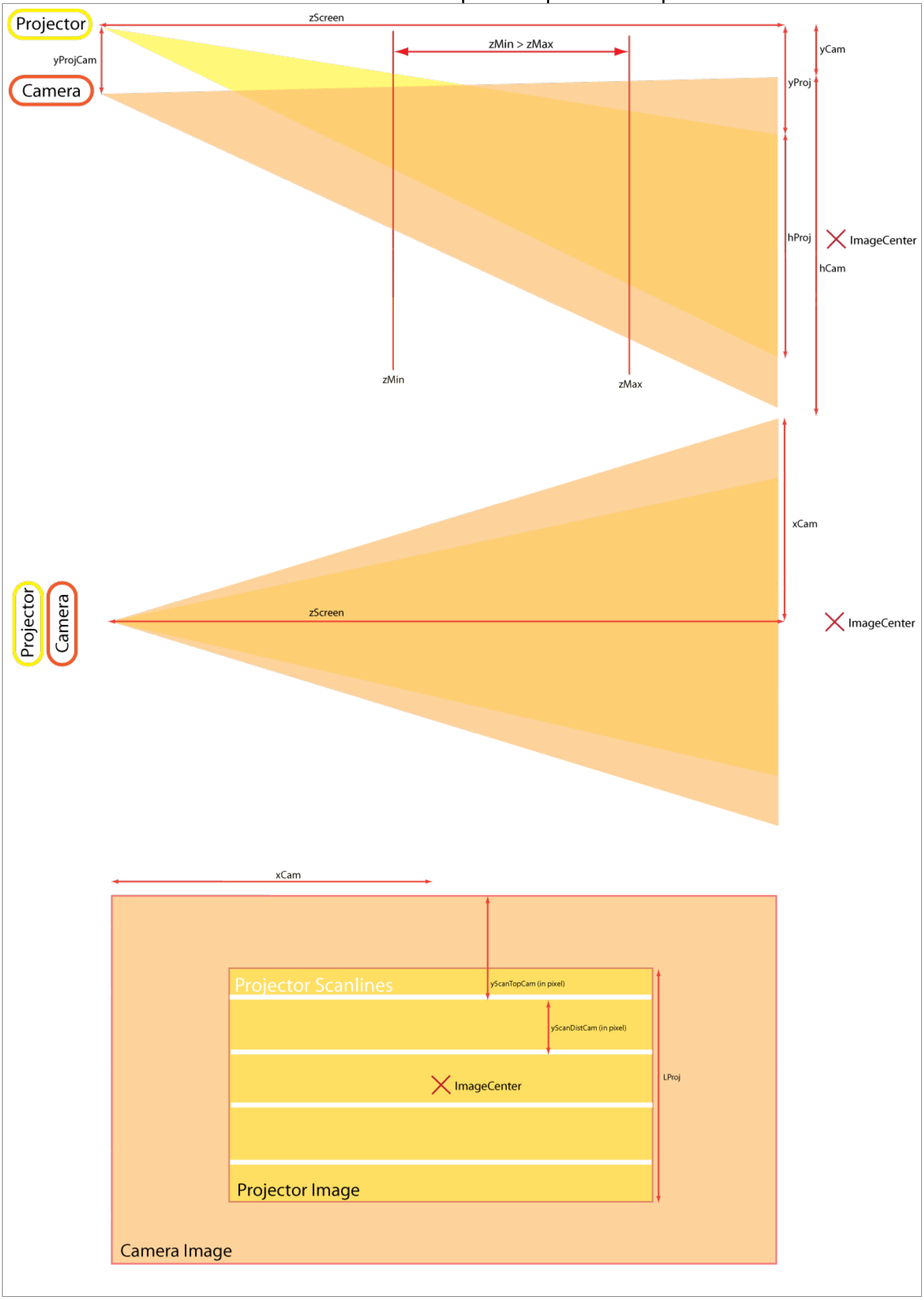
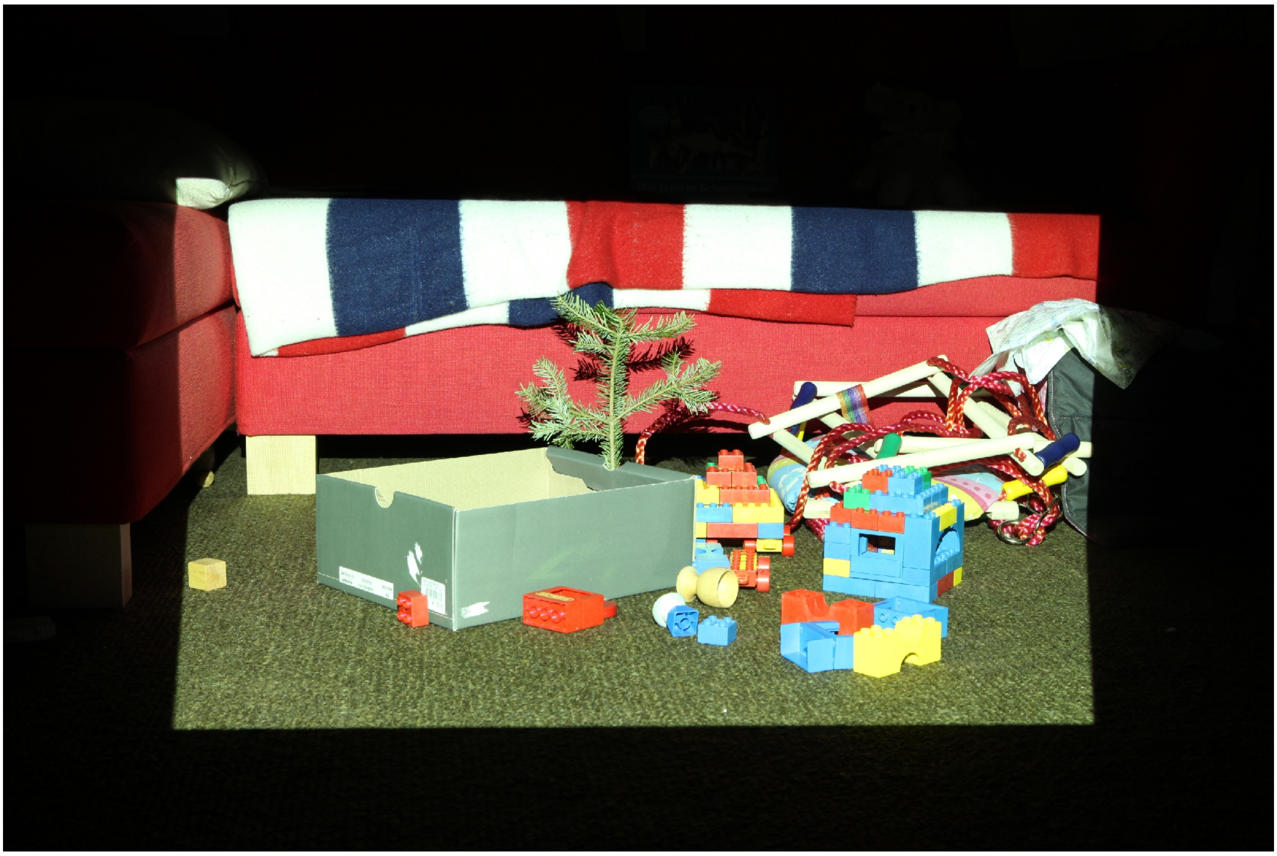
Sceamer

Digitizing a large objects with a camera and a Projector

Schematic Overview of the Principle Setup Side // Top // Front View

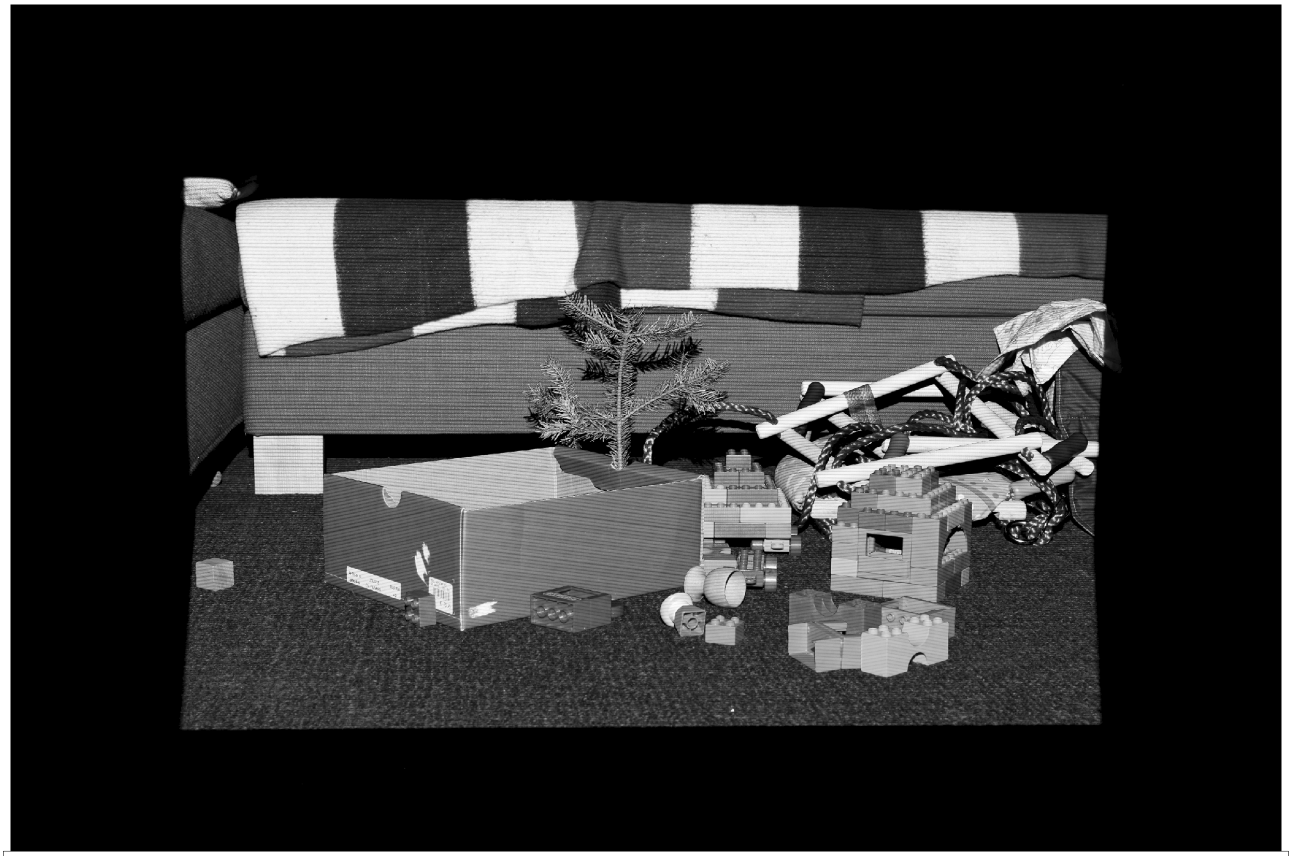




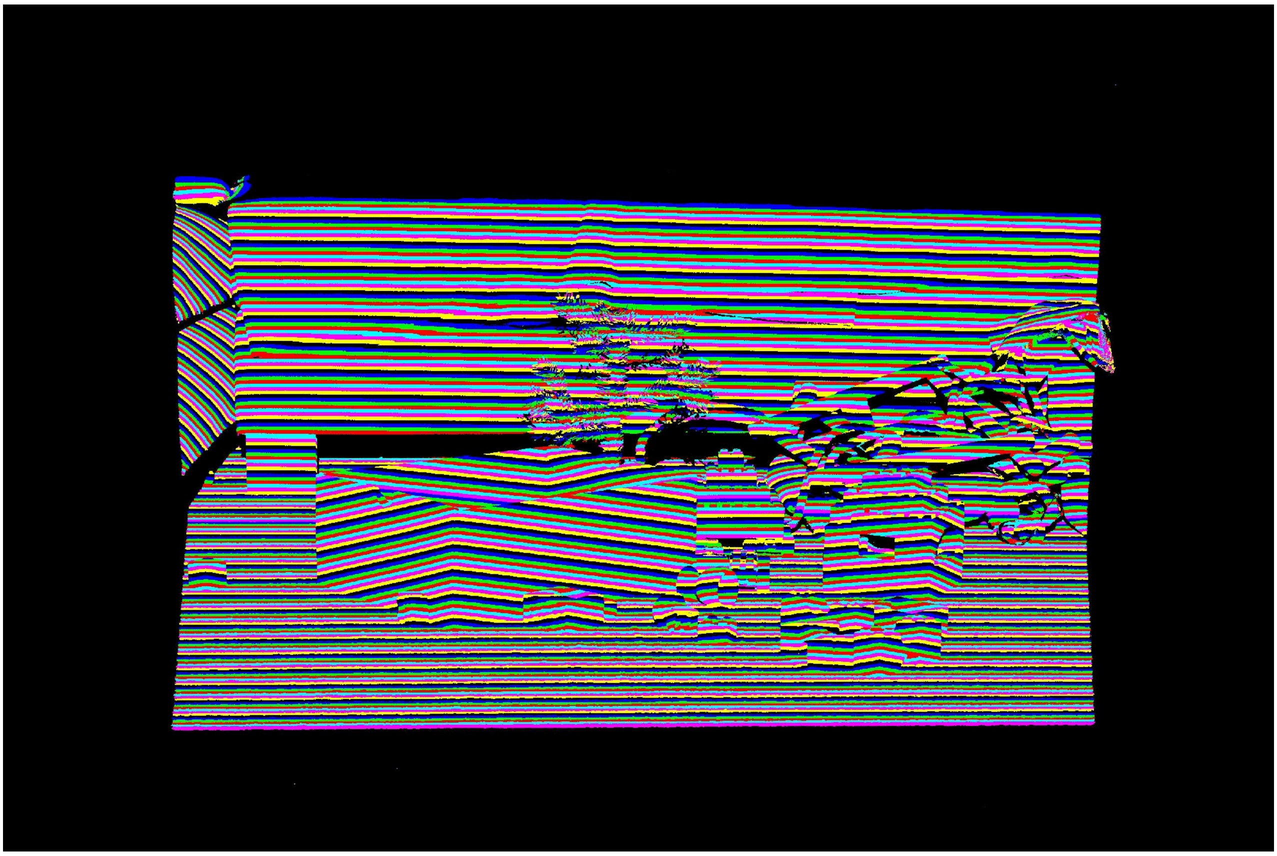
First Image „Scan_04_0001.JPG“

Second „Scan_04_0002.JPG“

Calculated Control Images

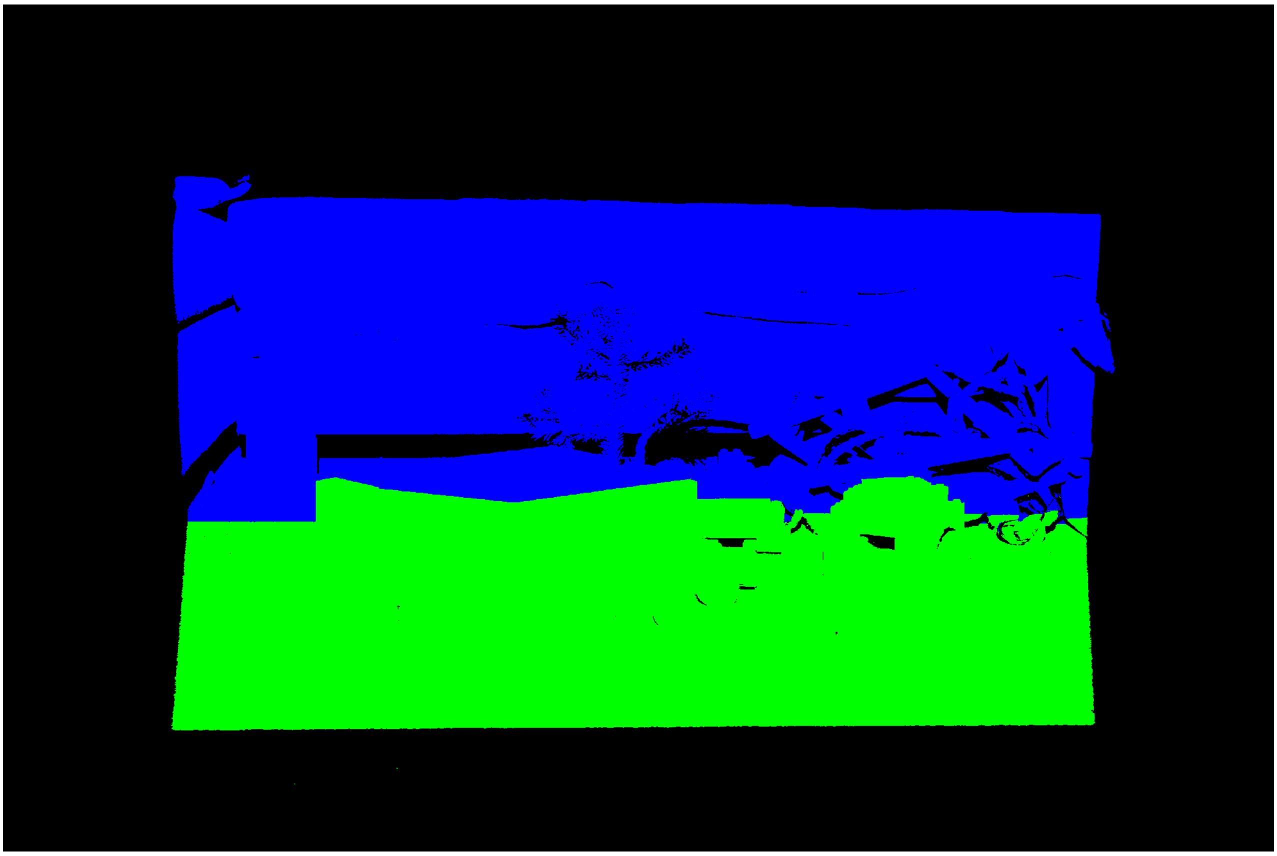


„Brightness.bmp“

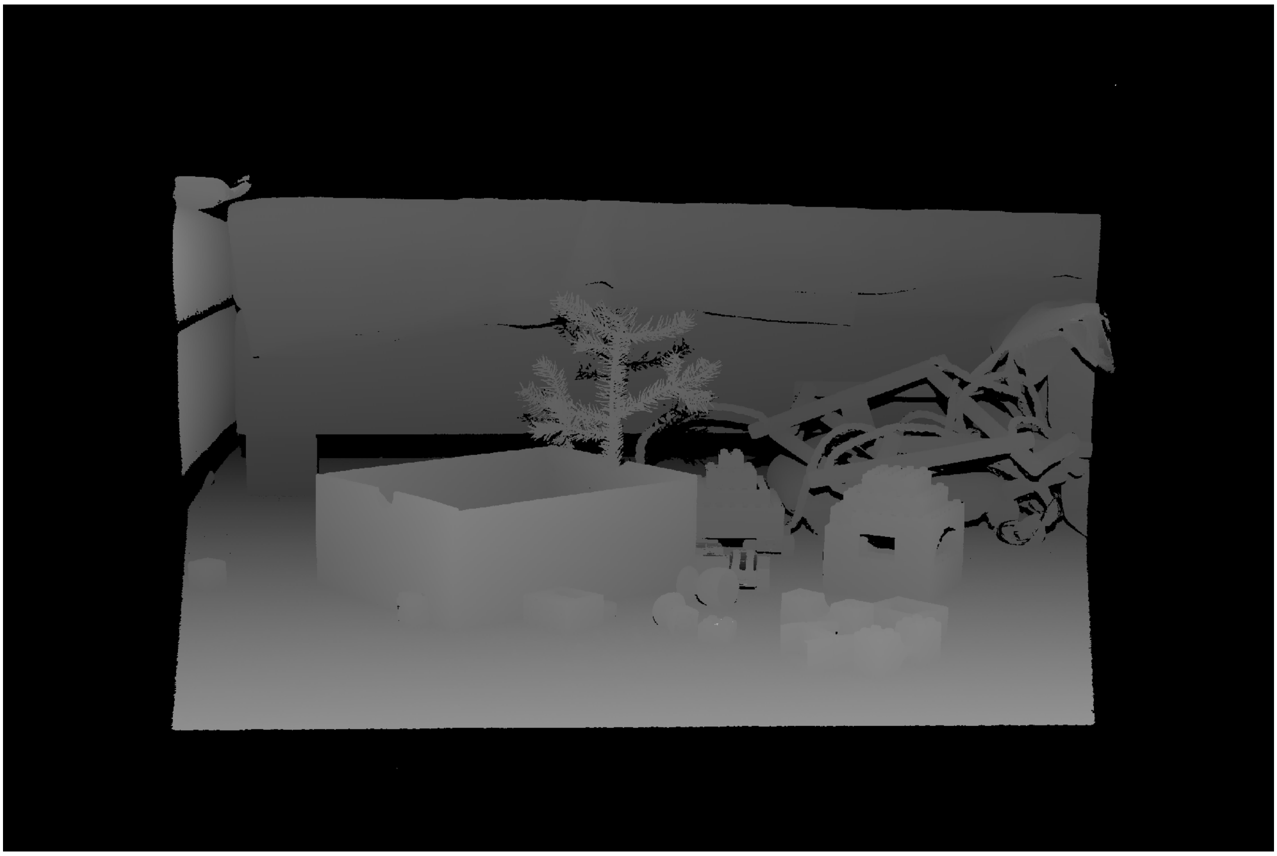


PictureNumberMap.bmp

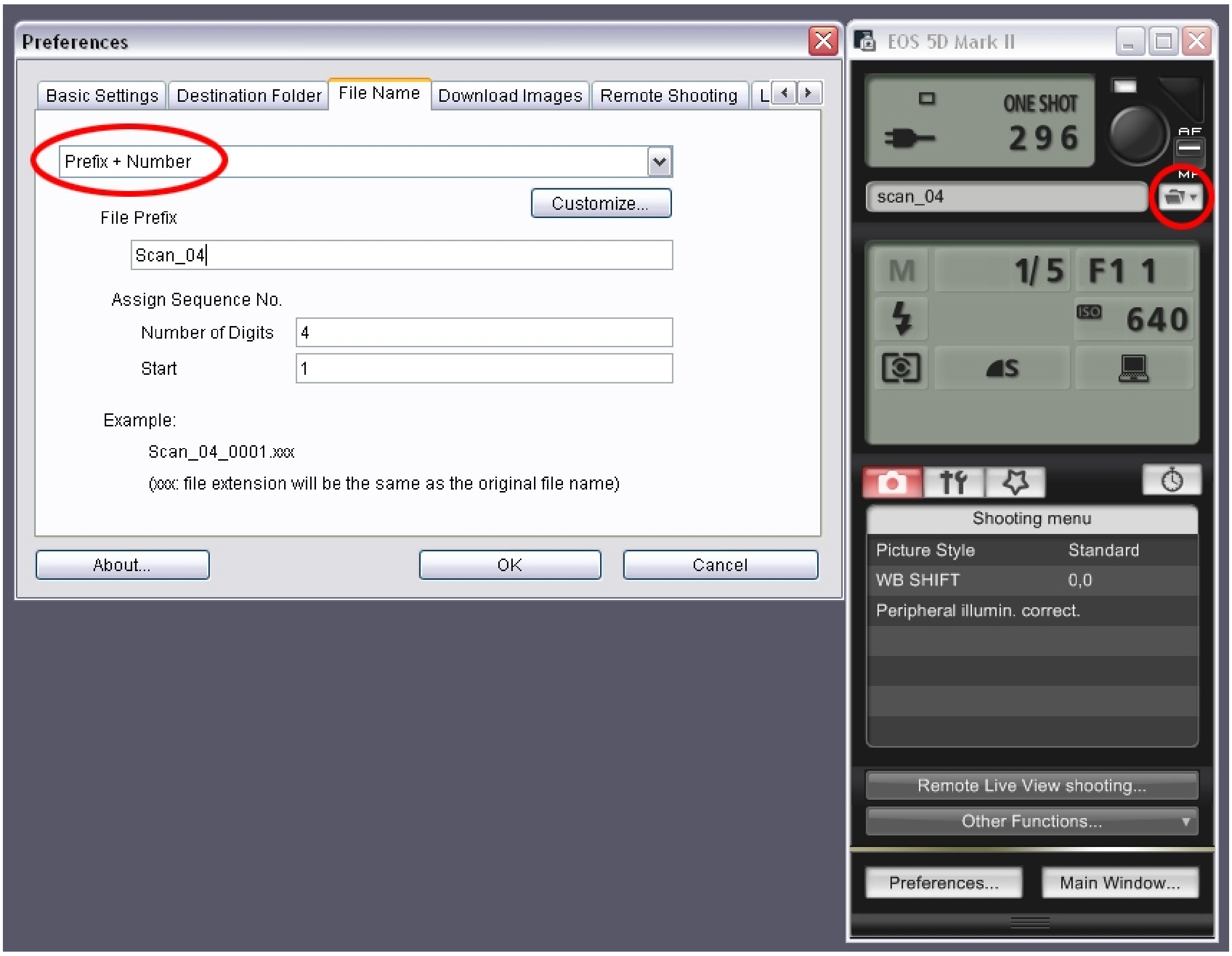
sg



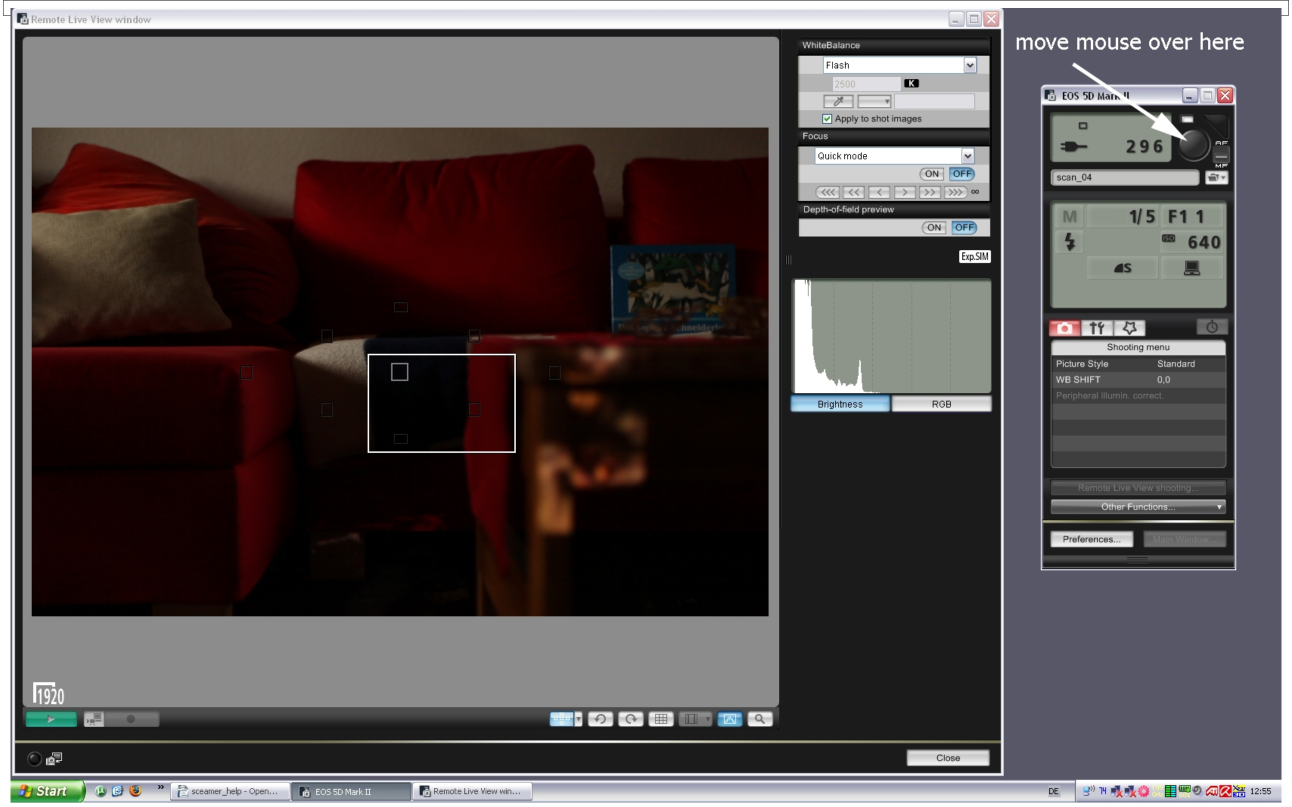
„BeamNumberMap.bmp“ (with only 2 Beams)



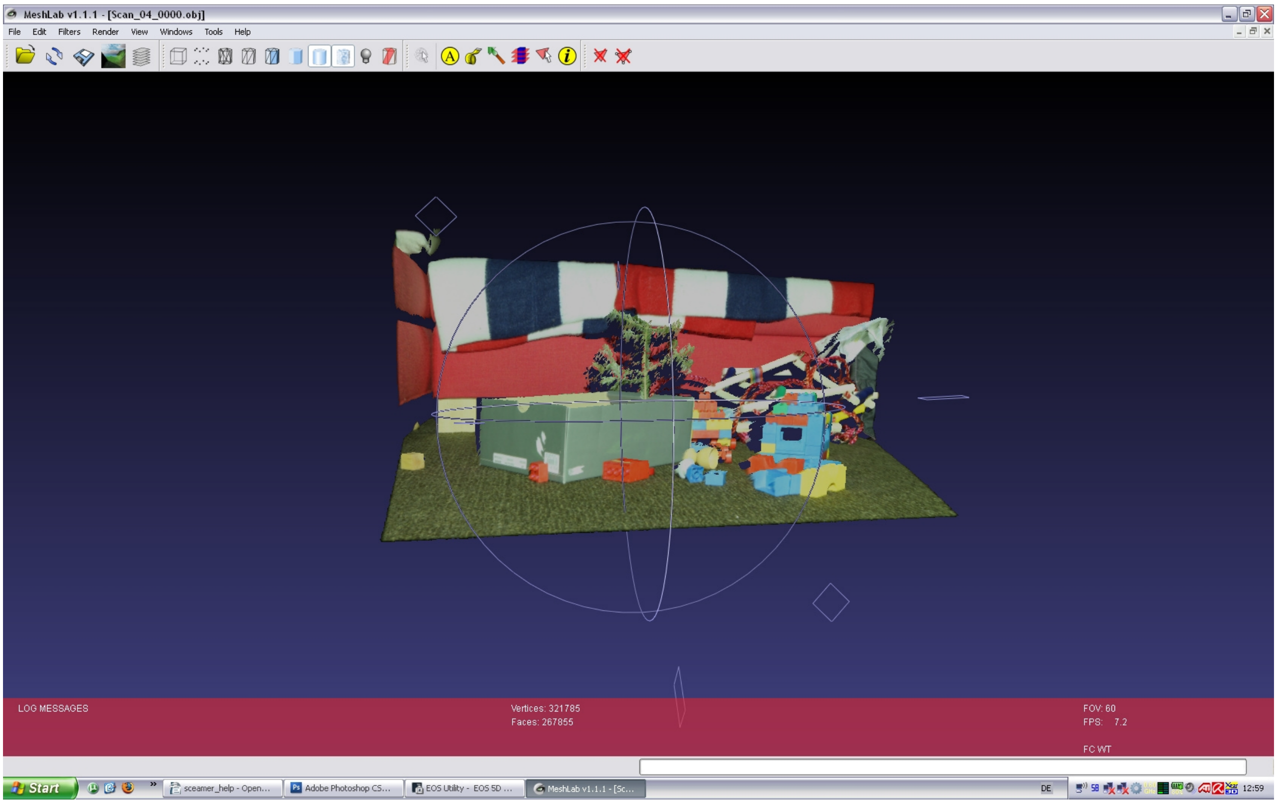
Resulting „DepthMap.bmp“



Eos Utility File saving settings

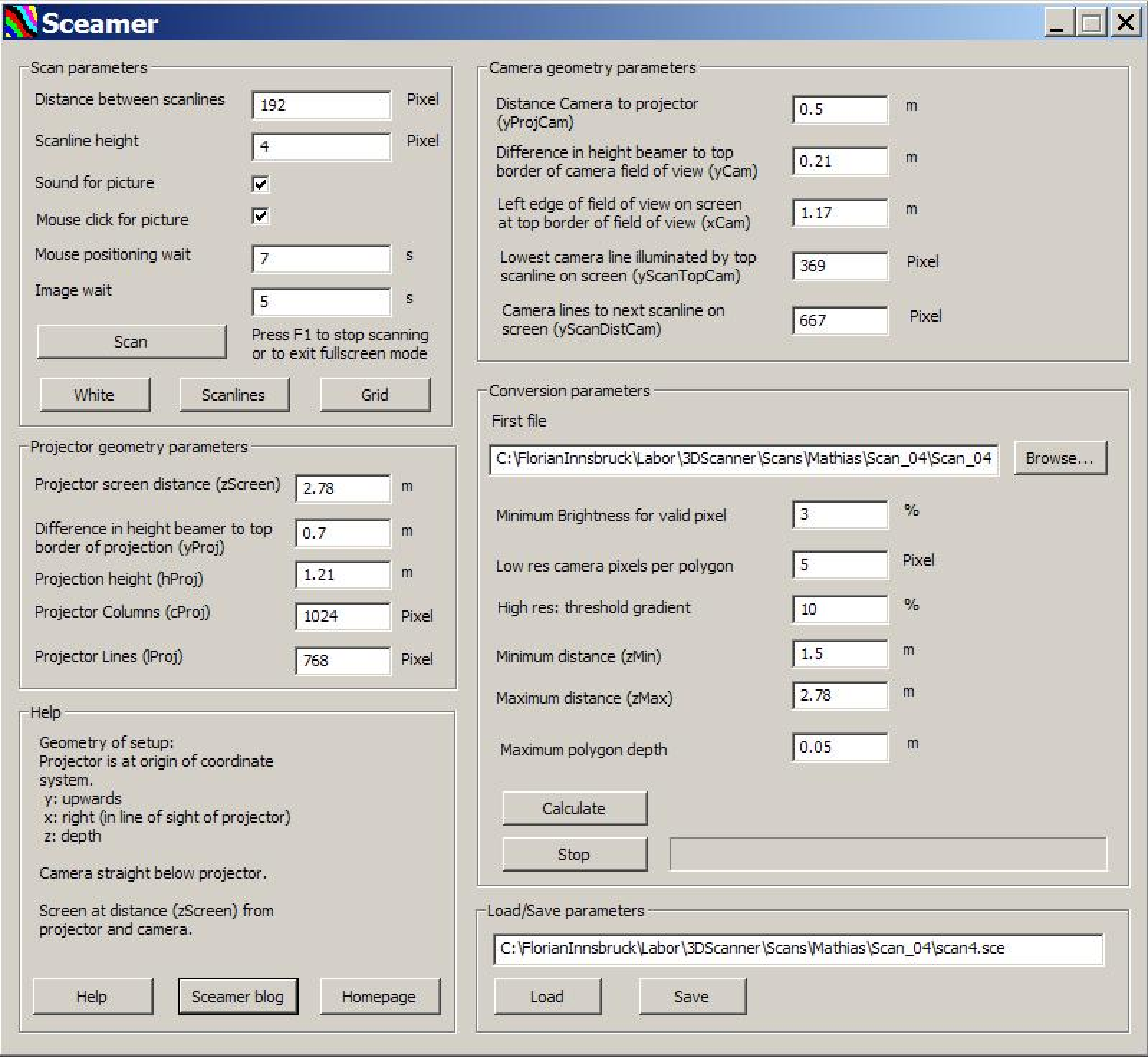


Eos Live View (Note Mouse Position for Sceamer Rec)



Viewing the Obj Mesh

http://meshlab.sourceforge.net/


Sceamer
Scan parameters
Distance between scanlines
192
Pixel
Scanline height
4
Pixel
Sound for picture
Mouse click for picture
Mouse positioning wait
7
s
Image wait
5
s
Scan
Press F1 to stop scanning or to exit fullscreen mode
White
Scanlines
Grid
Projector geometry parameters
Projector screen distance (zScreen)
2.78
m
Difference in height beamer to top border of projection (yProj)
0.7
m
Projection height (hProj)
1.21
m
Projector Columns (cProj)
1024
Pixel
Projector Lines (lProj)
768
Pixel
Help
Geometry of setup:
Projector is at origin of coordinate system.
y: upwards
x: right (in line of sight of projector)
z: depth
Camera straight below projector.
Screen at distance (zScreen) from projector and camera.
Help
Sceamer blog
Homepage
Camera geometry parameters
Distance Camera to projector (yProjCam)
0.5
m
Difference in height beamer to top border of camera field of view (yCam)
0.21
m
Left edge of field of view on screen at top border of field of view (xCam)
1.17
m
Lowest camera line illuminated by top scanline on screen (yScanTopCam)
369
Pixel
Camera lines to next scanline on screen (yScanDistCam)
667
Pixel
Conversion parameters
First file
C:\FlorianInnsbruck\Labor\3DScanner\Scans\Mathias\Scan_04\Scan_04
Browse...
Minimum Brightness for valid pixel
3
%
Low res camera pixels per polygon
5
Pixel
High res: threshold gradient
10
%
Minimum distance (zMin)
1.5
m
Maximum distance (zMax)
2.78
m
Maximum polygon depth
0.05
m
Calculate
Stop
Load/Save parameters
C:\FlorianInnsbruck\Labor\3DScanner\Scans\Mathias\Scan_04\scan4.sce
Load
Save

Calculation of object coordinates (see Figure 1)

Linear equation for illumination beam

$$yProj(z) = -\,yProjScreen \cdot \frac{z}{zScreen}$$

with $$yProjScreen = yProj + \underbrace{LineNumberProj \cdot \frac{hProj}{lProj}}_{\sigma Proj} \qquad (I)$$

Linear equation for „camera beam"

$$yCam(z) = -\,yProjCam + (yProjCam - yCamScreen) \cdot \frac{z}{zScreen}$$

with $$yCamScreen = yCam + \underbrace{LineNumberCam \cdot \frac{hCam}{lCam}}_{\sigma Cam} \qquad (II)\ (*)$$

Object at depth $zObj$, hit by both beams

$$yProj(zObj) = yCam(zObj) \equiv yObj \qquad (IV)$$

Solve for $zObj$

$$-\,yProjScreen\,\frac{zObj}{zScreen} = -\,yProjCam + (yProjCam - yCamScreen)\,\frac{zObj}{zScreen}$$

$$zObj\,\frac{yProjCam - yCamScreen + yProjScreen}{zScreen} = yProjCam$$

$$zObj = \frac{yProjCam \cdot zScreen}{yProjCam - yCamScreen + yProjScreen} \qquad (III)$$

Position in x Direction (see Figure 2)

linear equation for „camera beam"

$$xCam(z) = xCamScreen \cdot \frac{z}{zScreen}$$

with $$xCamScreen = xCam - ColumnNumberCam \cdot \frac{2\,xCam}{cCam} \qquad (V)\ (*)$$

$$xObj = xCam(zObj) = xCamScreen\,\frac{zObj}{zScreen} \qquad (VI)$$

$$\Rightarrow \quad \vec{r}\,Obj = \begin{pmatrix} xObj \\ yObj \\ zObj \end{pmatrix}$$

(*) only valid if camera orthogonal to screen
correct formulas for tilted camera: see below

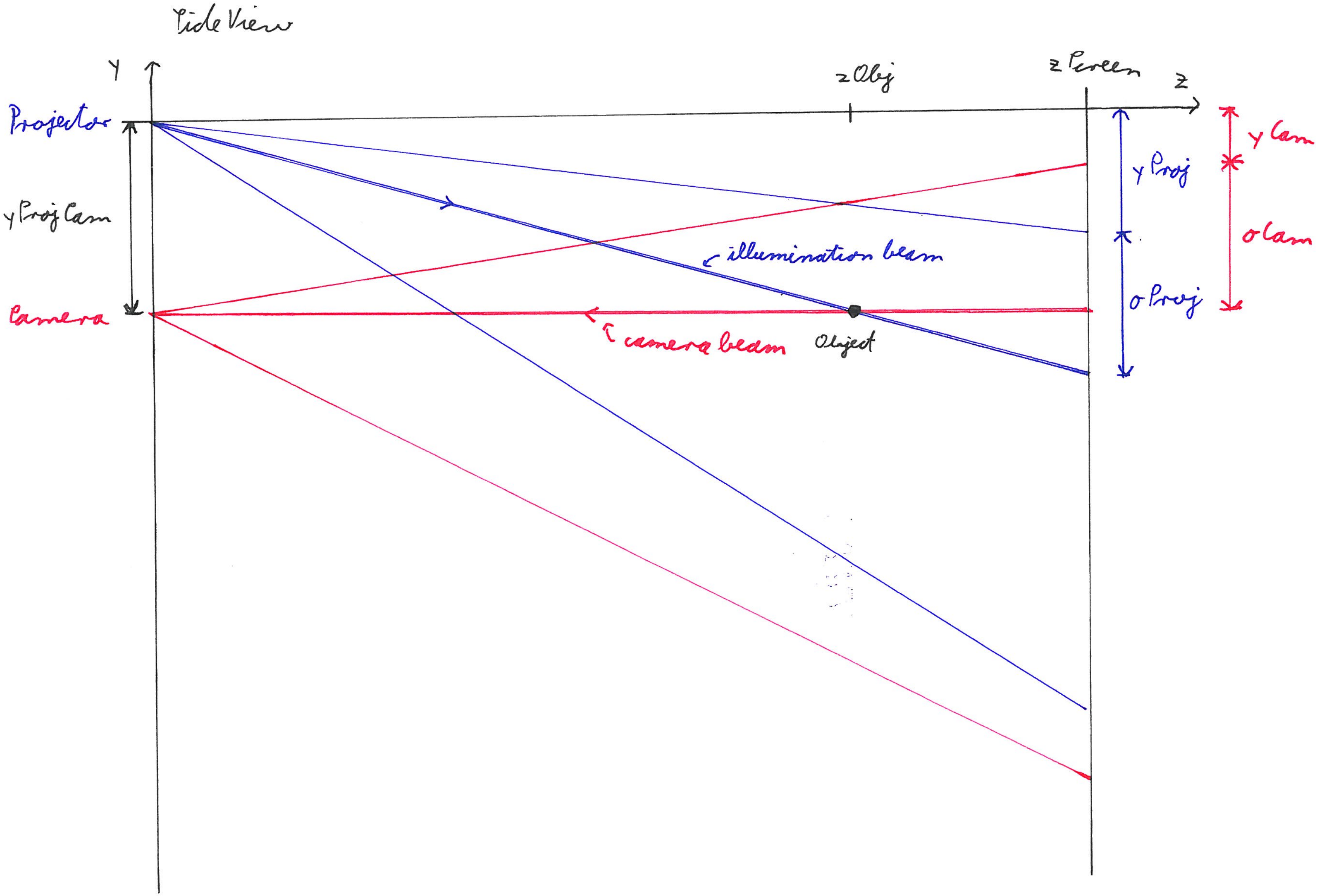

Side View
Y
z
zObj
zScreen
Projector
Camera
yProjCam
illumination beam
camera beam
Object
yProj
oProj
yCam
oCam

Figure 1

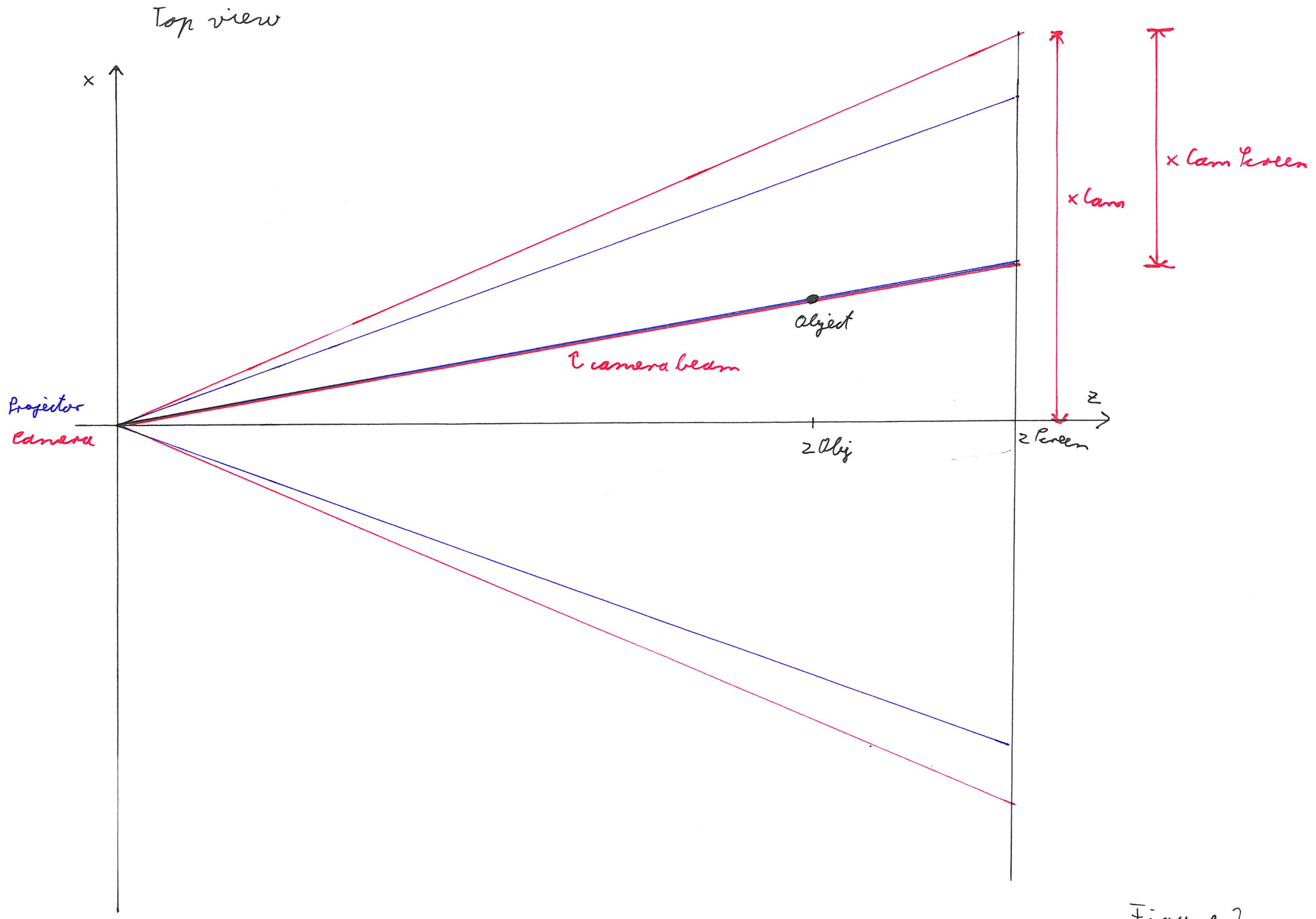

Top view
x
z
Projector
Camera
Object
z Obj
z Screen
x Cam
x Cam Screen

Figure 2

Conversion from camera coordinates to screen coordinates for tilted camera (see Figure 3)

$$l_1 = \sqrt{zScreen^2 + (yProjCam - yCam)^2}$$

$$l_3 = 2\,xCam \cdot \frac{lCam}{cCam}$$

$$\alpha_4 = atan\left(\frac{yProjCam - yCam}{zScreen}\right)$$

$$\alpha_2 = acos\left(\frac{l_3/2}{l_1}\right)$$

$$\alpha_3 = 90° - \alpha_4 - \alpha_2$$

$$\alpha_1 = 90° - \alpha_2$$

$$\alpha_5 = \alpha_3 + \alpha_1$$

$$l_4 = zScreen \cdot \tan(\alpha_5)$$

$$hCam = yProjCam - yCam + l_4$$

$$l_2 = \sqrt{zScreen^2 + l_4^2}$$

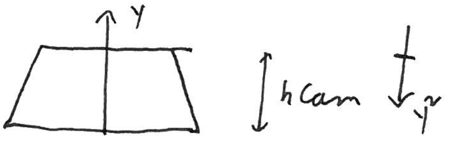
$$pCam = lineNumberCam \cdot \frac{l_3}{lCam}$$

$$oCam = pCam \cos\alpha_3$$

use this oCam in the formula for yCamScreen

Determination of $xCam(\tilde{y})$

Camera viewing area on screen:



$$\tilde{y} = yCam - y$$

Size on upper border: $2\,xCam(0)$

Size on lower border: $2\,xCam(0)\,\frac{l_2}{l_1}$

linear interpolation:

$$xCam(\tilde{y}) = xCam(0) + xCam(0)\left(\frac{l_2}{l_1} - 1\right)\frac{\tilde{y}}{hCam}$$

use this xCam in the formula for xCamScreen

Side view

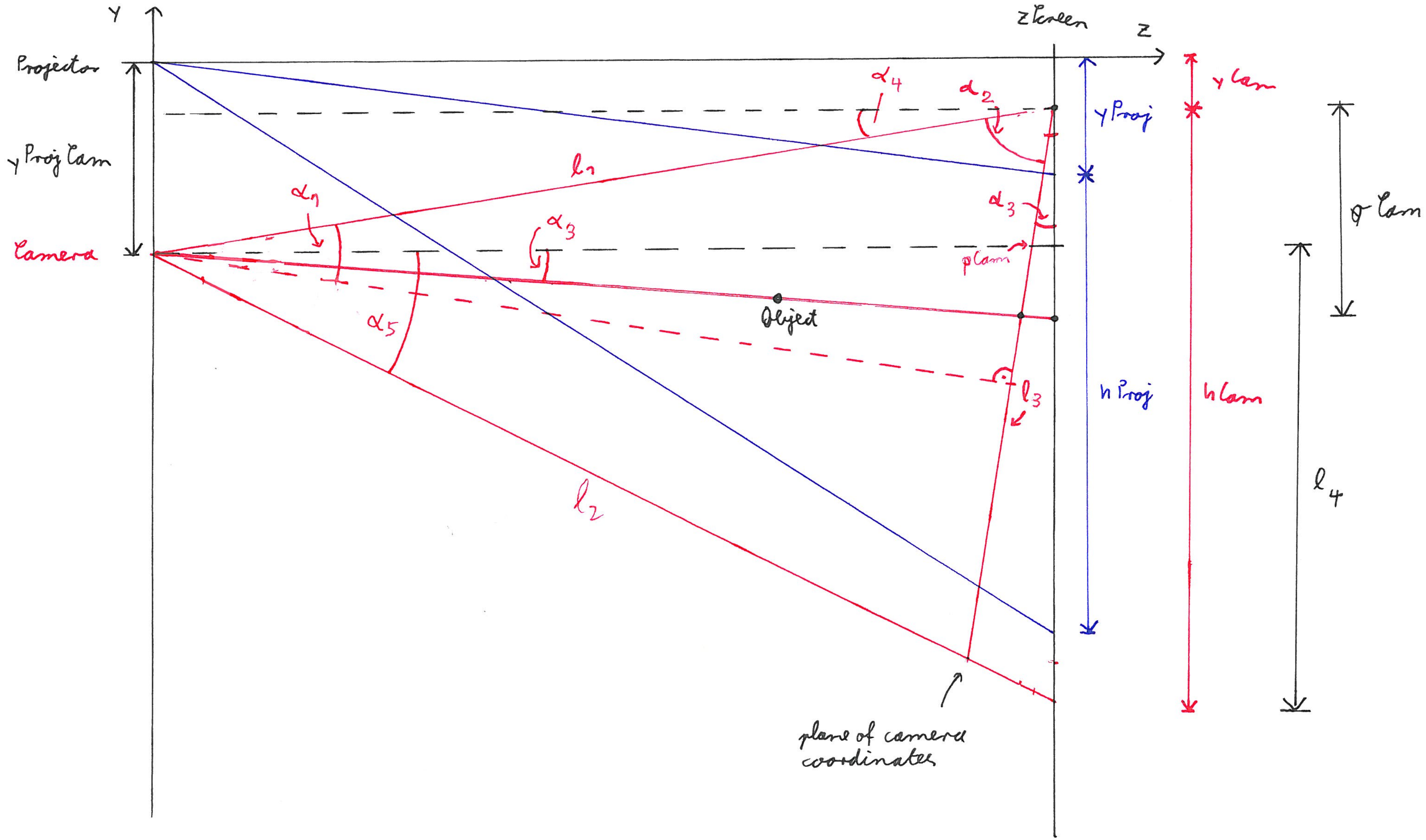


Figure 3

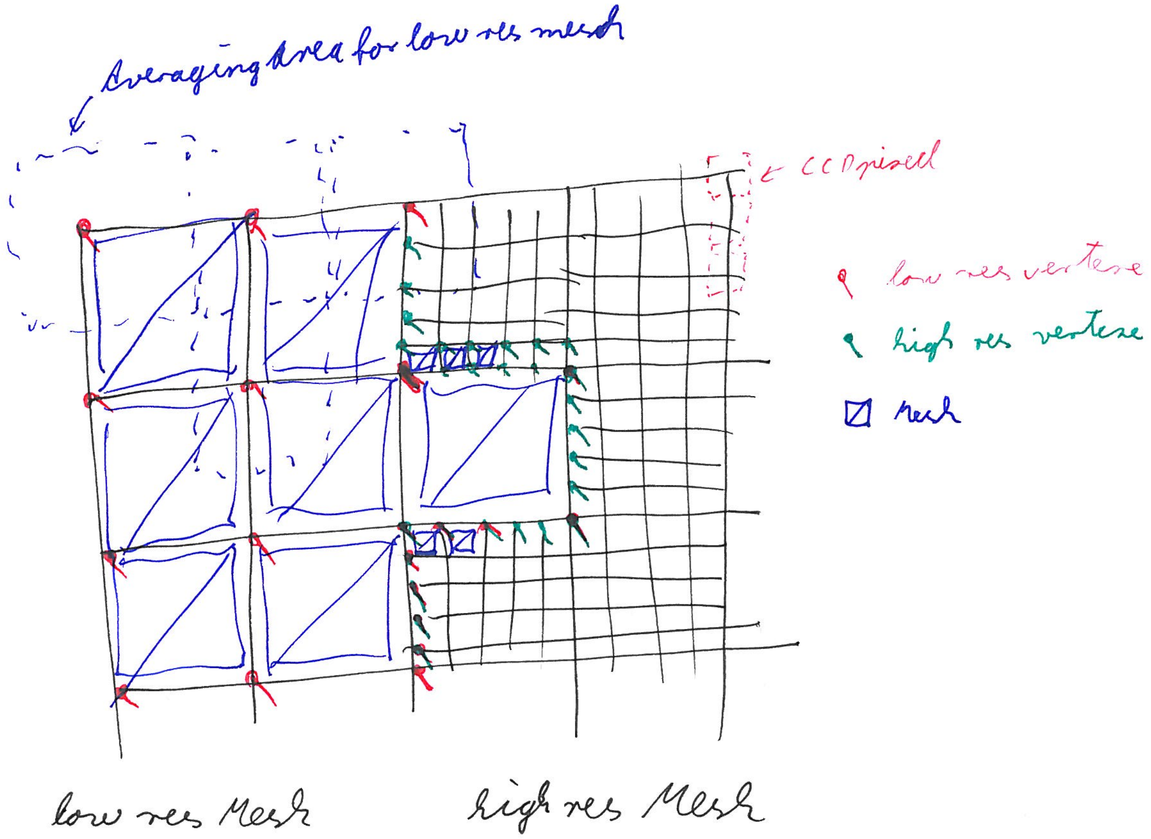

Averaging area for low res mesh
CCD pixel
low res vertexe
high res vertexe
mesh
low res Mesh
high res Mesh

Figure 4