

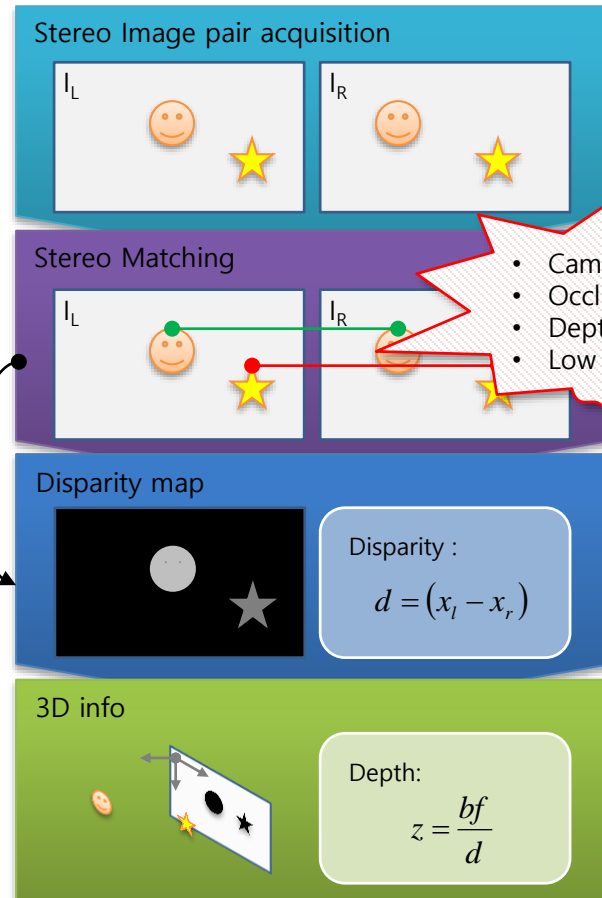
BumbleBee – 3D Stereo Camera

Jin-Hyung Kim

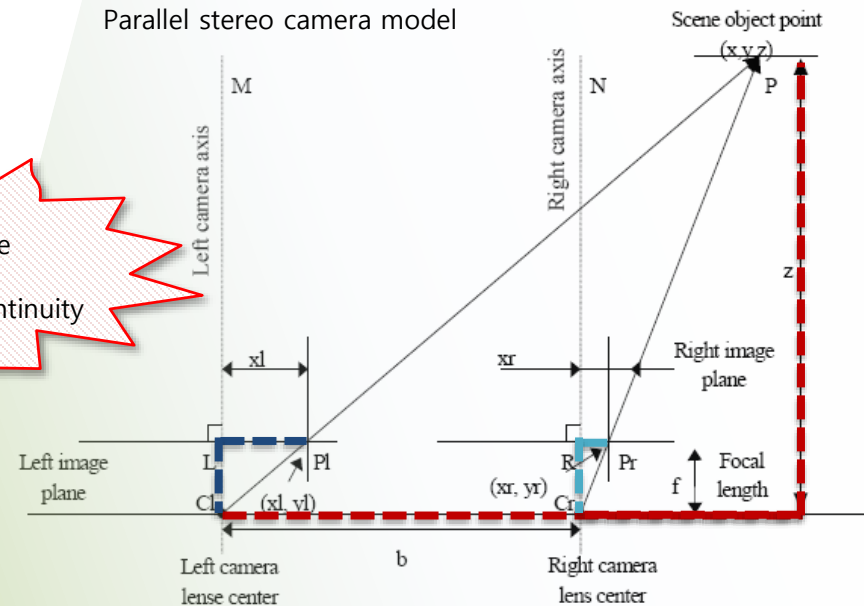
Lab seminar

Stereo Vision

● General Process & Principle



- Camera noise
- Occlusion
- Depth discontinuity
- Low texture



$$\frac{x}{z} = \frac{x_l}{f}, \quad \frac{x-b}{z} = \frac{x_r}{f}$$

$$z = \frac{bf}{(x_l - x_r)}$$

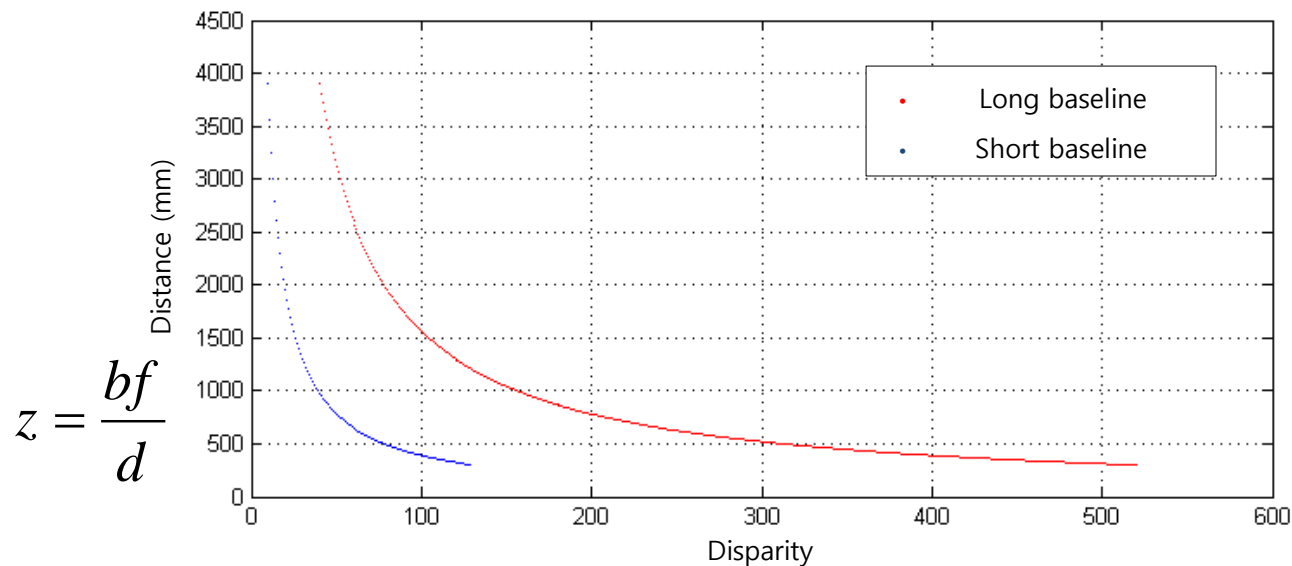
Tradeoff btw precision and accuracy

Long baseline

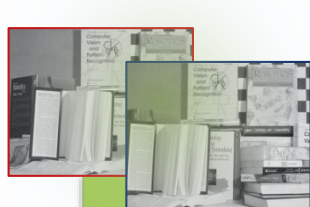
- Estimated distance is more precise
- Longer disparity range causes
 - Heavy computation
 - High false matching possibility

Short baseline

- Estimated distance is less precise
- Shorter disparity range causes
 - Light computation
 - Low false matching possibility

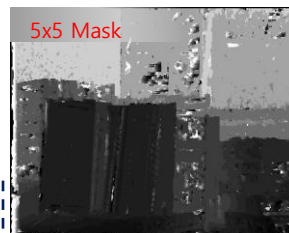
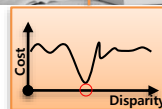
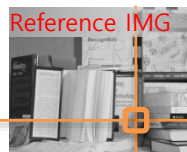


Conventional matching algorithm



Local

- Mask operation
- Cost function
- Local minima



$$SAD_{MN}(x, y, d) = \sum_{y=1}^M \sum_{x=1}^N |I_l(x, y) - I_r(x - d, y)|$$

$$SSD_{MN}(x, y, d) = \sum_{y=1}^M \sum_{x=1}^N [I_l(x, y) - I_r(x - d, y)]^2$$

$$MAE_{MN}(x, y, d) = \frac{1}{M \times N} \sum_{y=1}^M \sum_{x=1}^N |I_l(x, y) - I_r(x - d, y)|$$

Global

- Energy based
- Global Priority
- Belief Propagation, Graph cut...



- Smooth depth connectivity
- Segmentation
- Pattern

$$E(f) = E_{data}(f) + \lambda E_{smooth}(f)$$

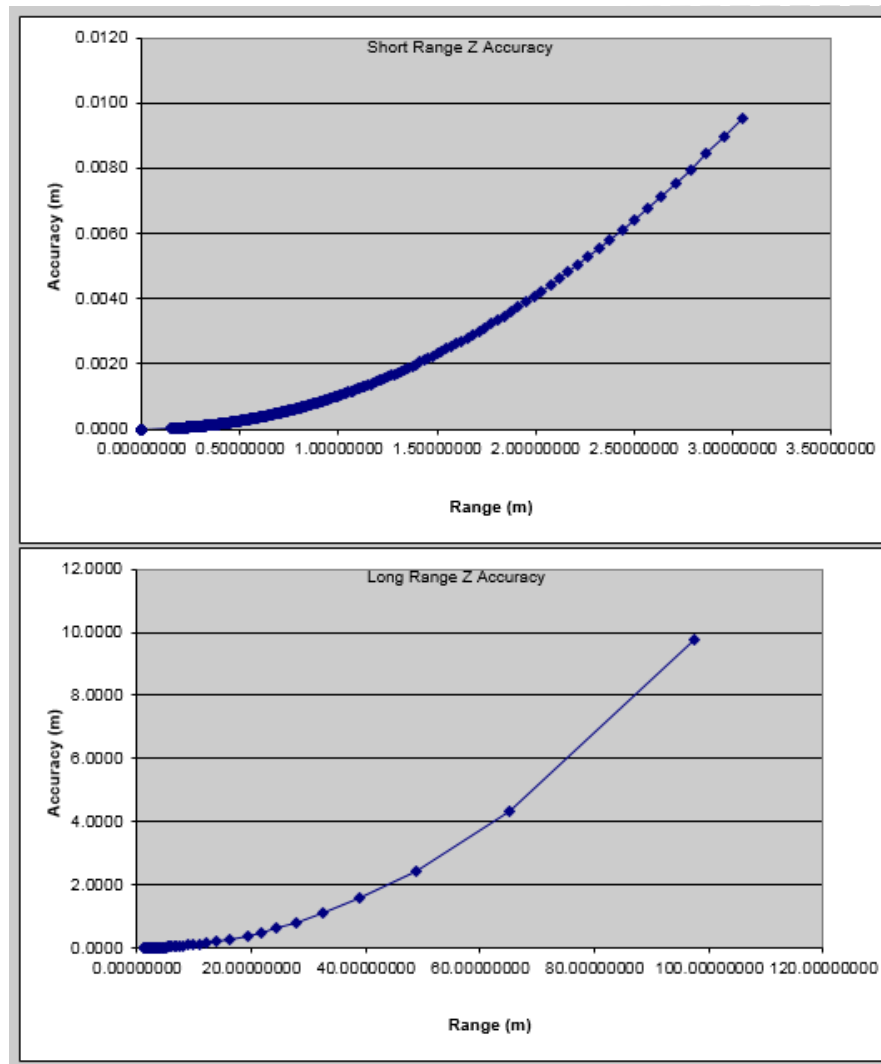
BumbleBee (Legacy)

- Digital stereo vision camera

Resolution	640 x 480
Sensor	Sony ICX204 Color CCD
Frame rate	30Hz
Baseline	12 cm
Weight	375 g
Interface	IEEE-1394a
Power consumption	2.1 W
Lens	6 mm (43 deg H-FOV)



Accuracy of BumbleBee

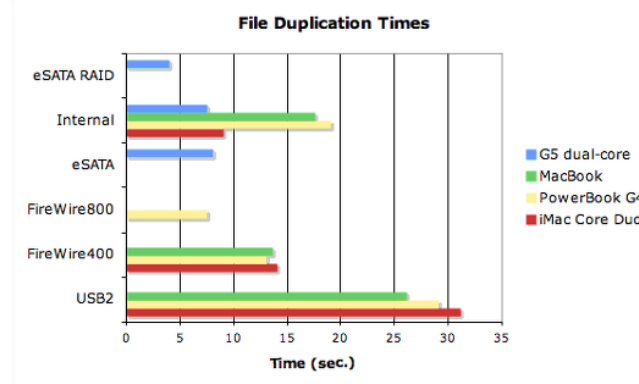


IEEE 1394 – Firewire, i.Link

- Apple이 제창한 개인용 컴퓨터 및 디지털 오디오, 디지털 비디오용

시리얼 버스 인터페이스 표준 규격

- 디지털 인터페이스 표준 : 반복적인 DAC에 발생하는 지속적인 신호의 감쇠를 처리
- 빠르고 용이한 전환 : SCSI와 달리 장치의 설치/제거가 컴퓨터 구동 중에도 용이
- 편리성 : PnP, 빠른 속도
- 4.5 m maximum
- 규격
 - IEEE-1394a : 100/200/400 Mbps
 - IEEE-1394b : 800Mbps
 - IEEE-1394c : 800Mbps



Setup Development environment

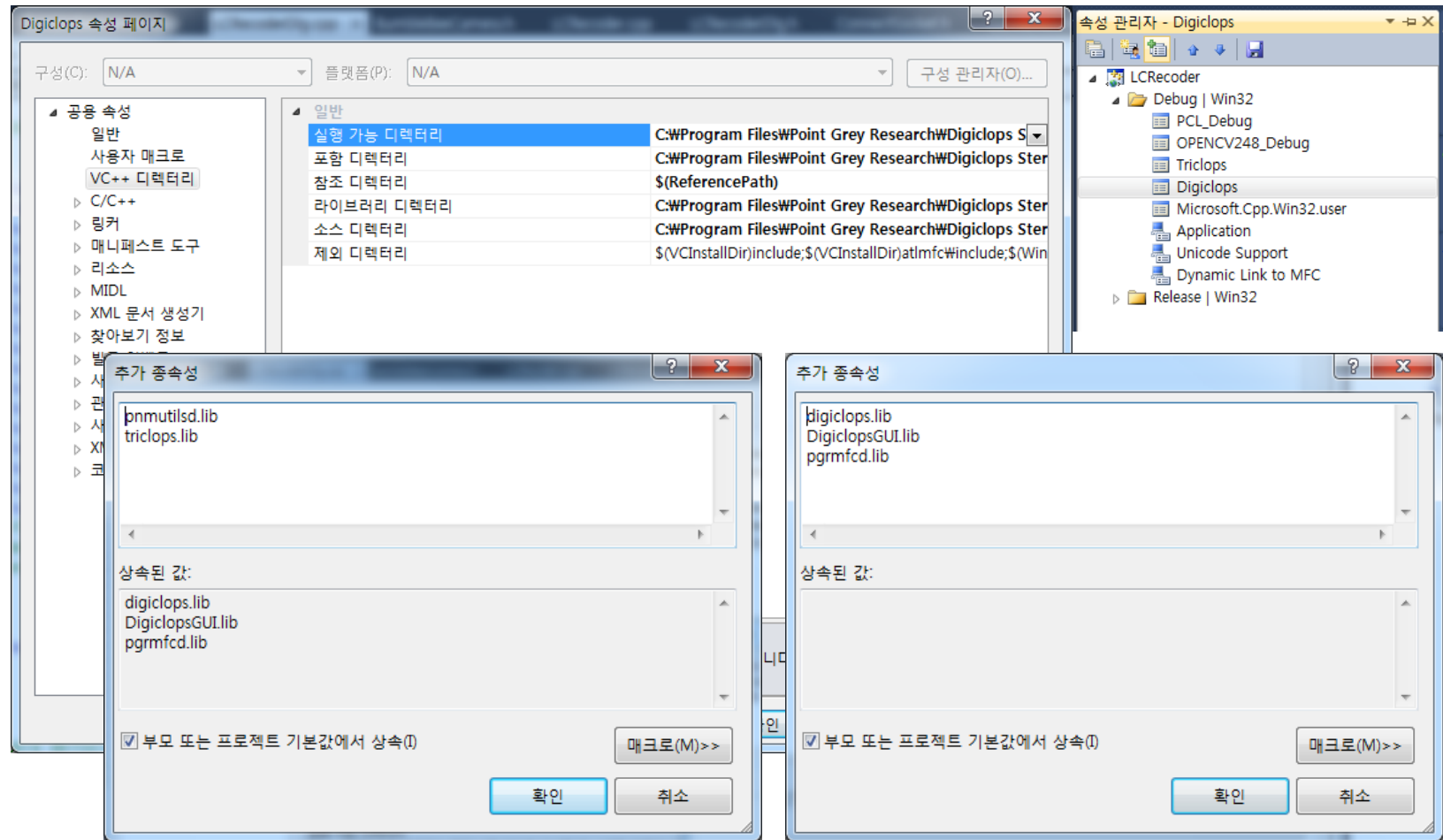
- What you need

- H/W
 - Bumblebee camera
 - IEEE-1394a Interface board / additional power for mobile
 - IEEE-1394a Cable
- S/W
 - Digiclops SDK (3.2b09)
 - Triclops SDK (2.5b05)



Setup Development environment

● Visual Studio

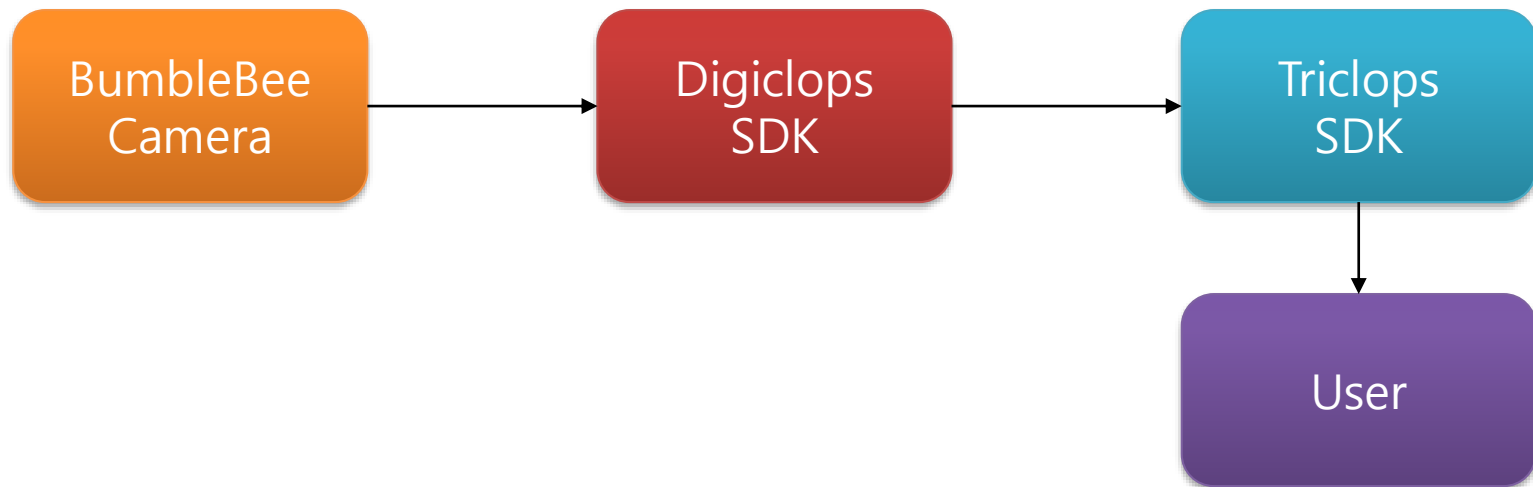


How to use SDK

- Include

```
#include <digiclops.h>  
#include <triclops.h>
```

- Work Flow



Initialization

```
( digiclopsCreateContext( &m_digiclopsContext ) );
( digiclopsInitialize( m_digiclopsContext, 0 ) );
( digiclopsGetTriclopsContextFromCamera( m_digiclopsContext, &m_triclopsContext ) );
( digiclopsGetCameraInfoEx( m_digiclopsContext, &m_info ) );
( digiclopsSetImageTypes( m_digiclopsContext, STEREO_IMAGE ) );|
( digiclopsSetImageResolution( m_digiclopsContext, DIGICLOPS_FULL ) );
( digiclopsSetFrameRate( m_digiclopsContext, DIGICLOPS_FRAMERATE_100 ) );
( digiclopsStart( m_digiclopsContext ) );

( triclopsSetResolutionAndPrepare(m_triclopsContext,h,w,
    imageHeight(),imageWidth()) );
( triclopsSetImageBuffer(m_triclopsContext,m_leftImage,
    TriImg_RECTIFIED,TriCam_LEFT) );
( triclopsSetImageBuffer(m_triclopsContext,m_rightImage,
    TriImg_RECTIFIED,TriCam_RIGHT) );

( triclopsSetImageBuffer(m_triclopsContext,left,
    TriImg_RECTIFIED,TriCam_LEFT) );
( triclopsSetImageBuffer(m_triclopsContext,right,
    TriImg_RECTIFIED,TriCam_RIGHT) );

triclopsGetFocalLength(m_triclopsContext, &fc);
triclopsGetImageCenter(m_triclopsContext, &cy, &cx);

triclopsSetDisparity(m_triclopsContext, 30, 80);
triclopsSetUniquenessValidationMapping(m_triclopsContext, 0);
triclopsSetUniquenessValidation(m_triclopsContext, 1);
triclopsSetTextureValidationMapping(m_triclopsContext, 0);
triclopsSetTextureValidation(m_triclopsContext, 1);
triclopsSetBackForthValidation(m_triclopsContext, 1);
triclopsSetBackForthValidationMapping(m_triclopsContext, 0);
triclopsSetSurfaceValidationMapping(m_triclopsContext, 0);
triclopsSetSurfaceValidation(m_triclopsContext, 1);
triclopsSetSubpixelValidationMapping(m_triclopsContext, 0);
```

Capture & Download

```
digiclopsGrabImage( m_digiclopsContext )
```



```
TriclopsInput inputStereo, inputLeft, inputRight;
TriclopsPackedColorImage outputLeft, outputRight;
( digiclopsExtractTriclopsInput(m_digiclopsContext, STEREO_IMAGE, &inputStereo ) );
if(m_hasColor) {
    ( digiclopsExtractTriclopsInput(m_digiclopsContext, LEFT_IMAGE, &inputLeft ) );
    ( digiclopsExtractTriclopsInput(m_digiclopsContext, RIGHT_IMAGE, &inputRight ) );
}

( triclopsRectify(m_triclopsContext, &inputStereo) );
if(m_hasColor) {
    ( triclopsRectifyPackedColorImage(m_triclopsContext, TriCam_LEFT,
    &inputLeft, &outputLeft) );
    ( triclopsRectifyPackedColorImage(m_triclopsContext, TriCam_RIGHT,
    &inputRight, &outputRight) );
}

triclopsPreprocess(m_triclopsContext, &inputStereo);
triclopsStereo(m_triclopsContext);

triclopsGetImage(m_triclopsContext, TriImg_DISPARIITY, TriCam_REFERENCE, &m_depthImage);
```

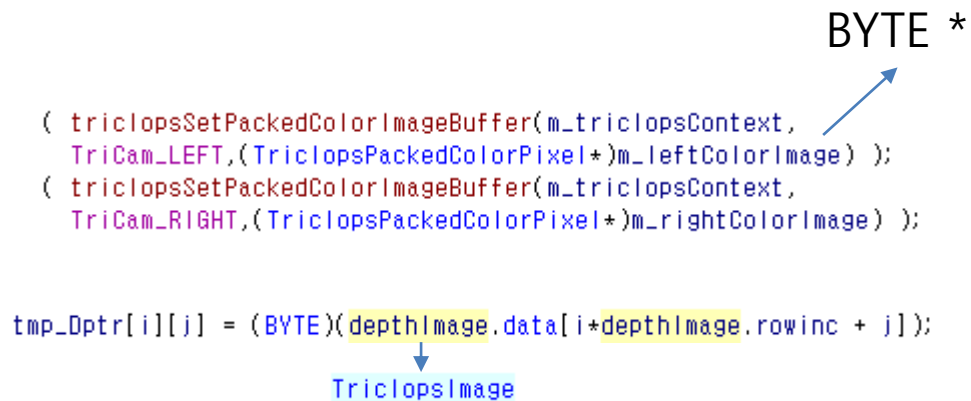
Set Buffer

BYTE *

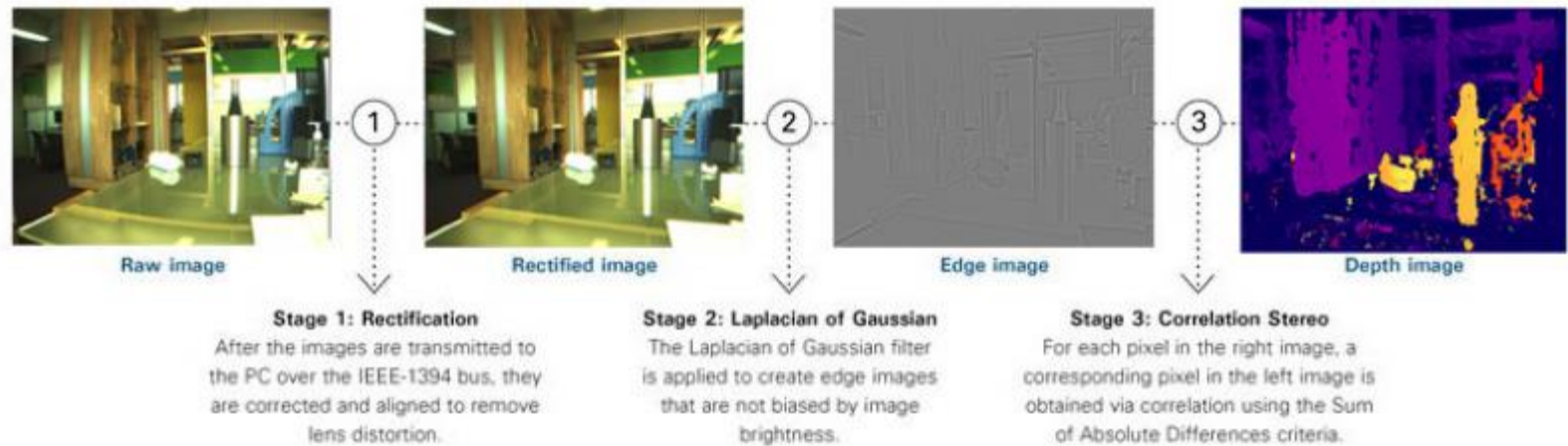
```
( triclopsSetPackedColorImageBuffer(m_triclopsContext,  
  TriCam_LEFT,(TriclopsPackedColorPixel*)m_leftColorImage) );  
( triclopsSetPackedColorImageBuffer(m_triclopsContext,  
  TriCam_RIGHT,(TriclopsPackedColorPixel*)m_rightColorImage) );
```

tmp_Dptr[i][j] = (BYTE)(depthImage.data[i*depthImage.rowinc + j]);

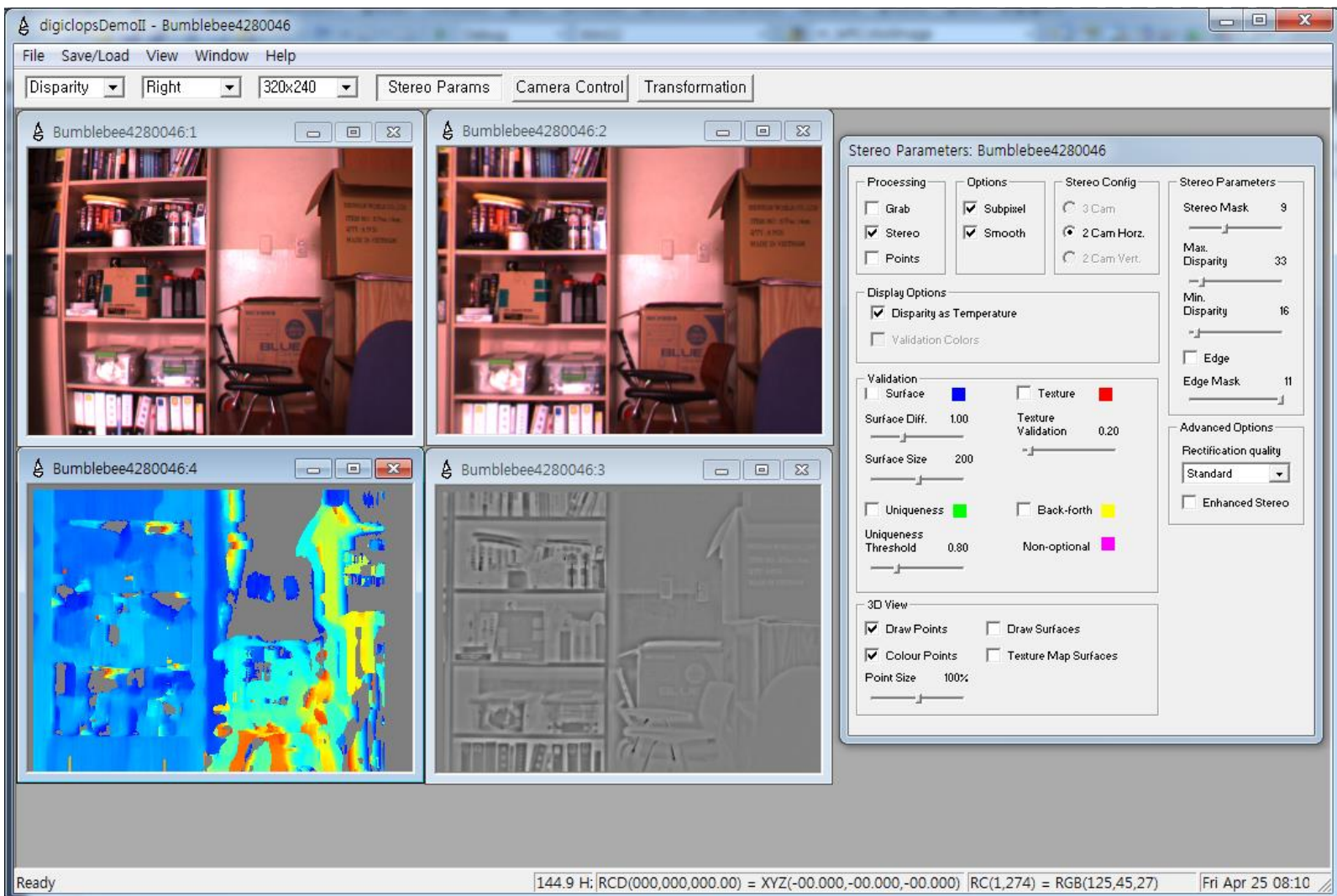
TriclopsImage



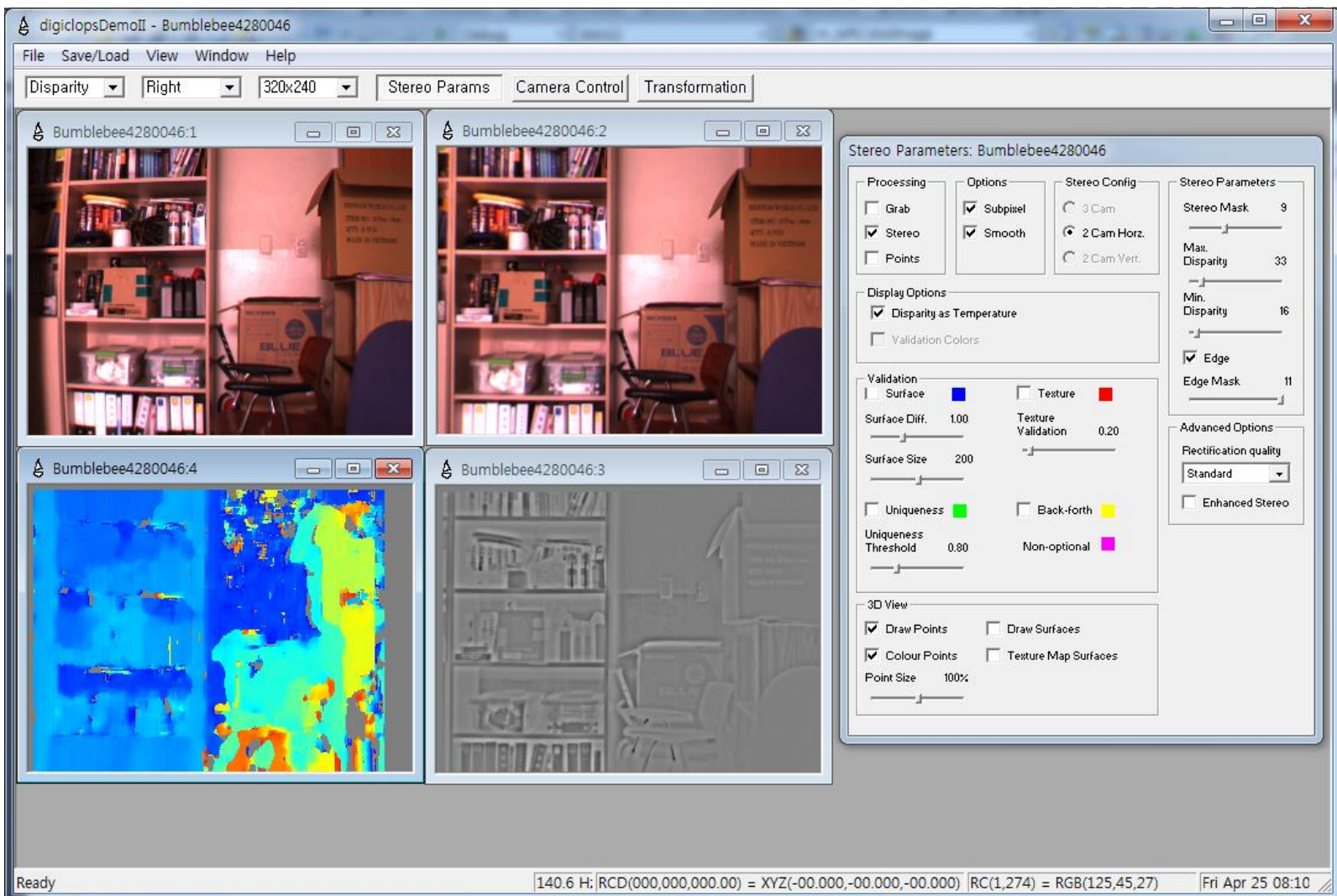
Bumblebee process



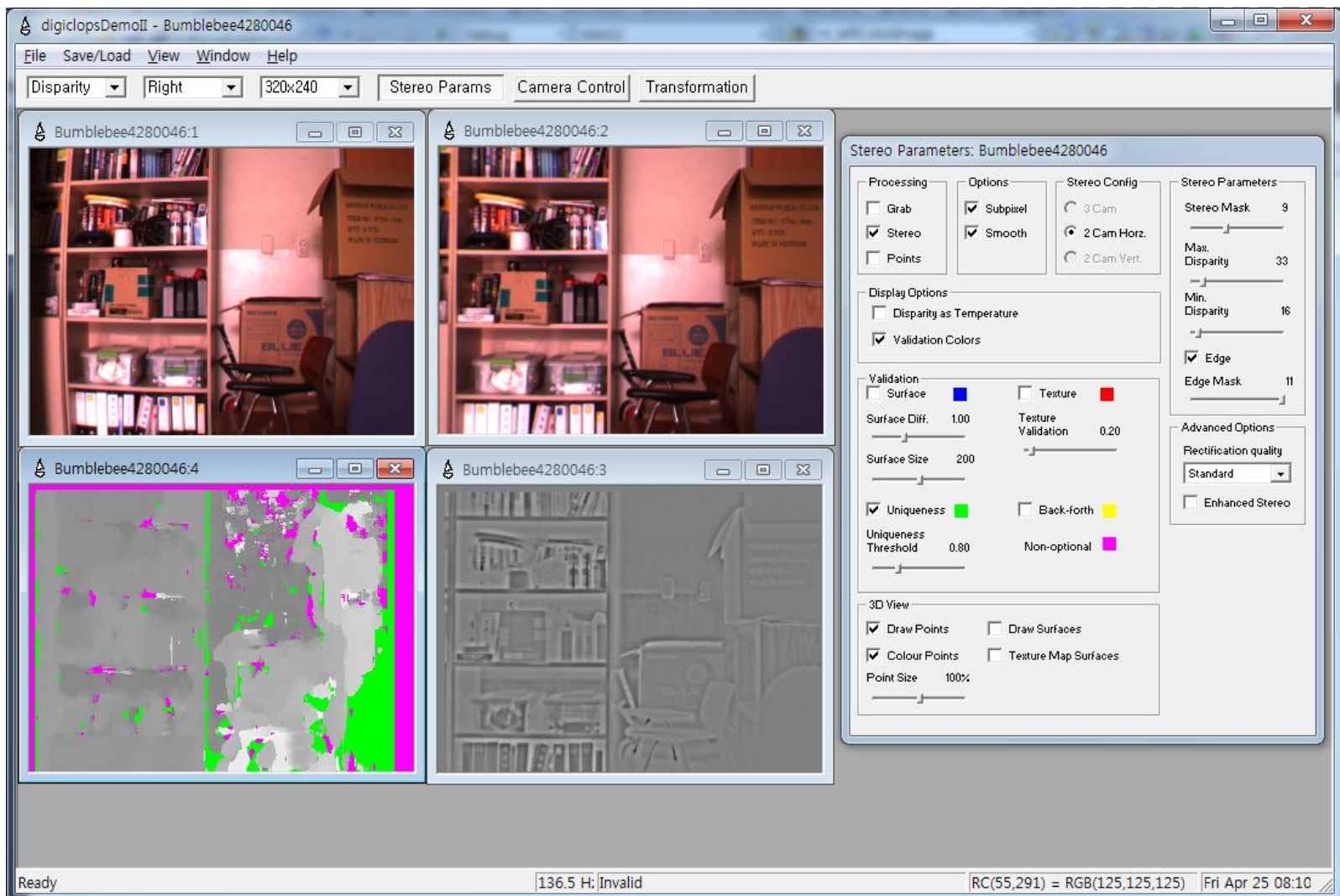
Disparity Validation – Init.



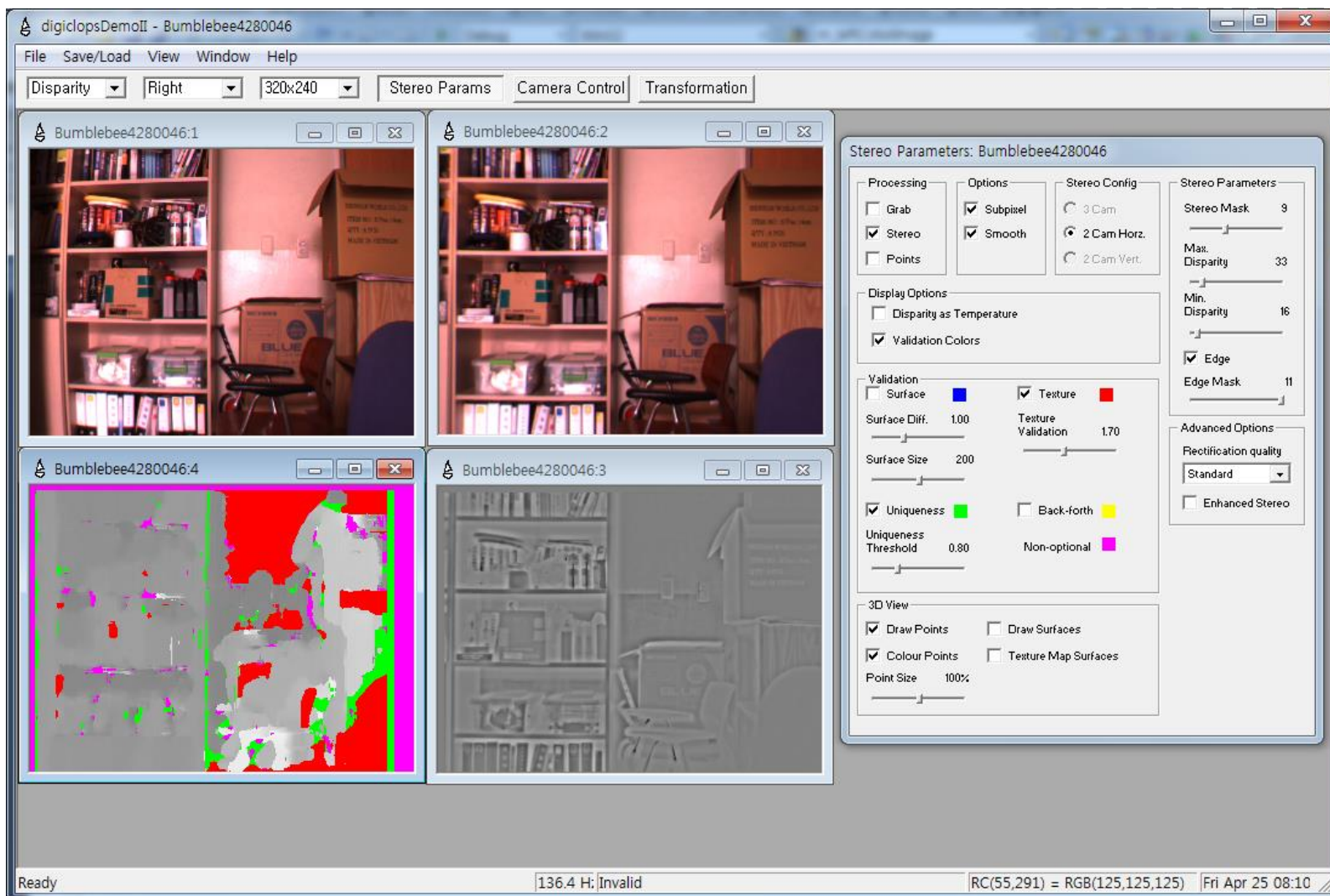
Disparity Validation - Edge



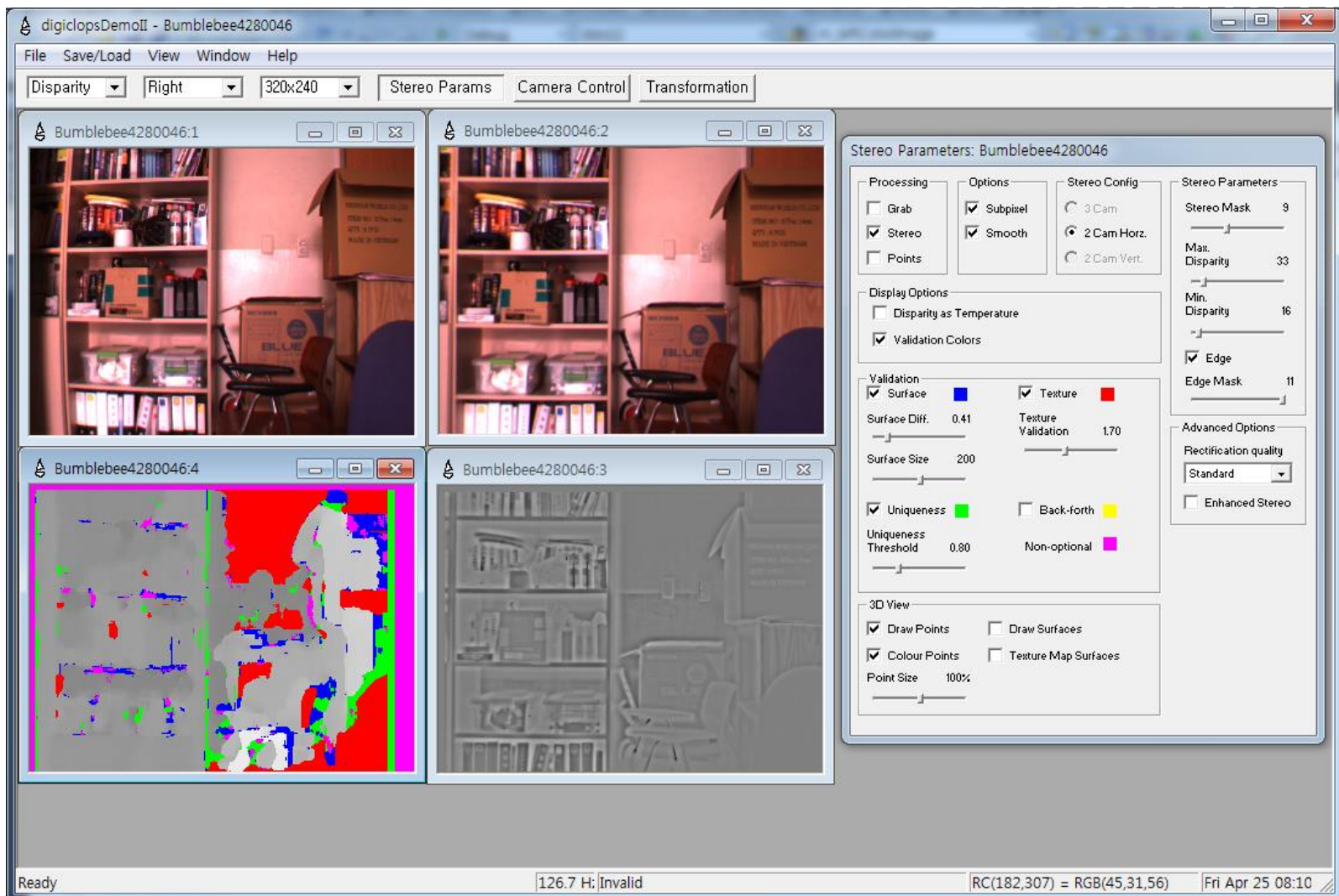
Disparity Validation - Uniqueness



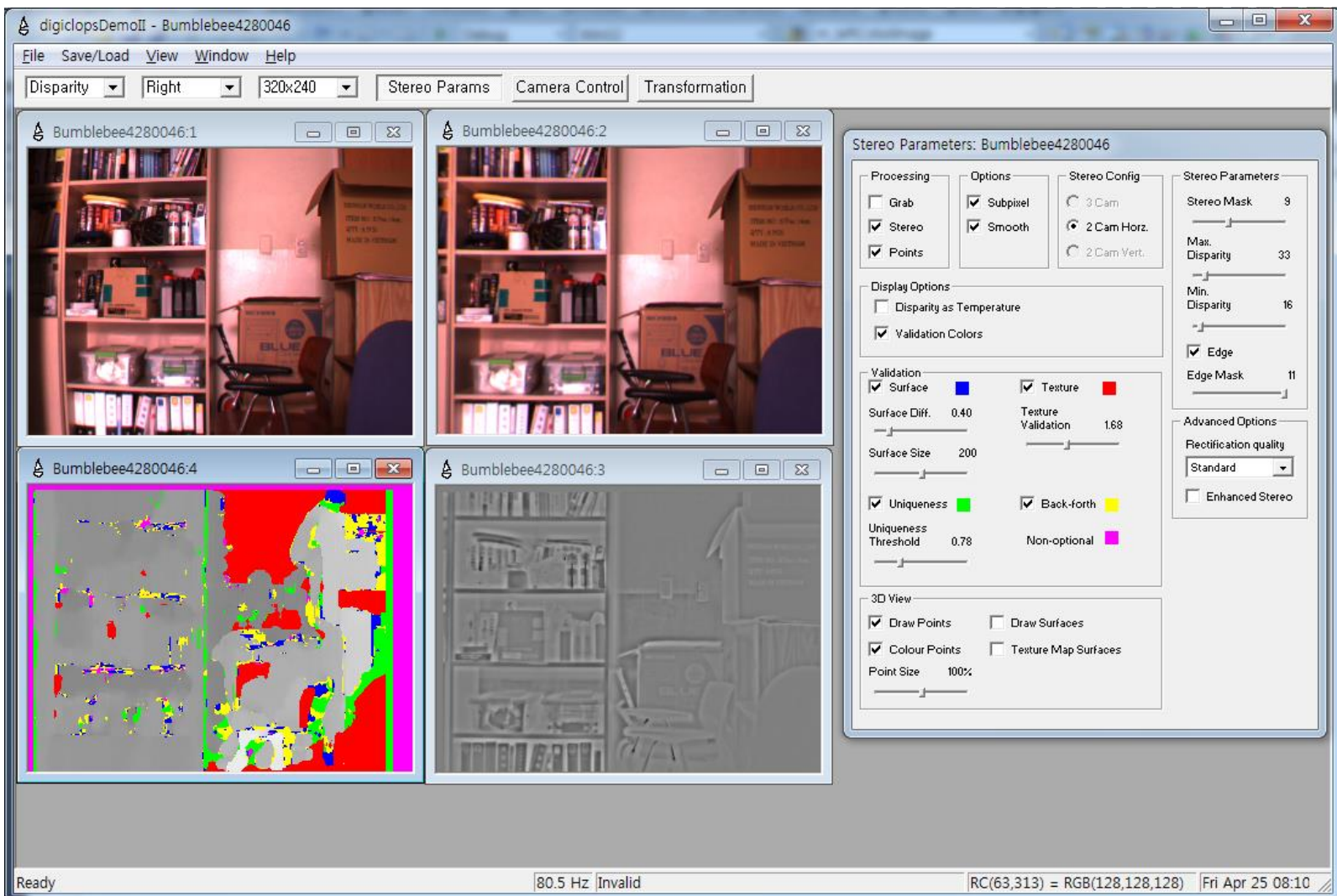
Disparity Validation - Texture



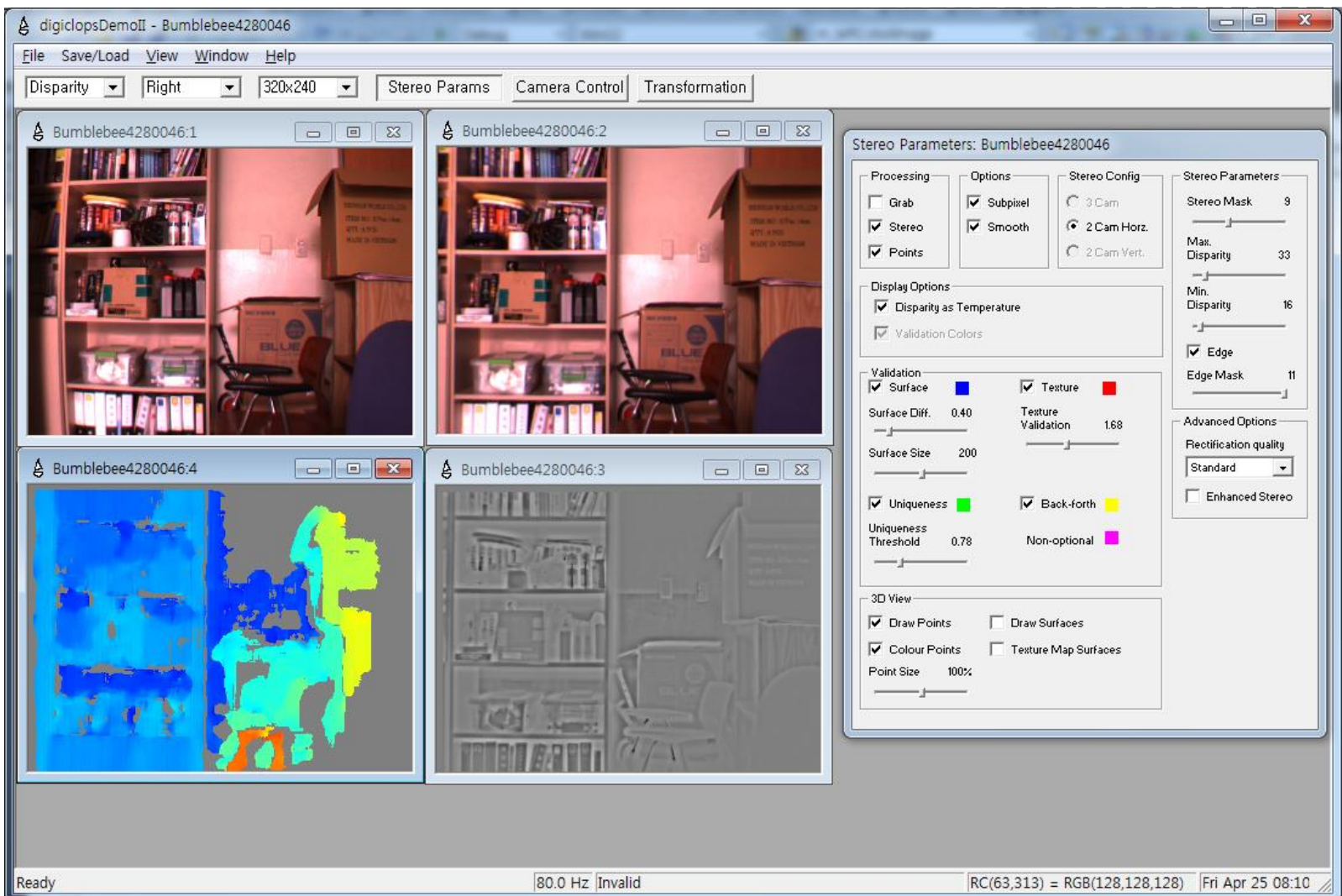
Disparity Validation - Surface



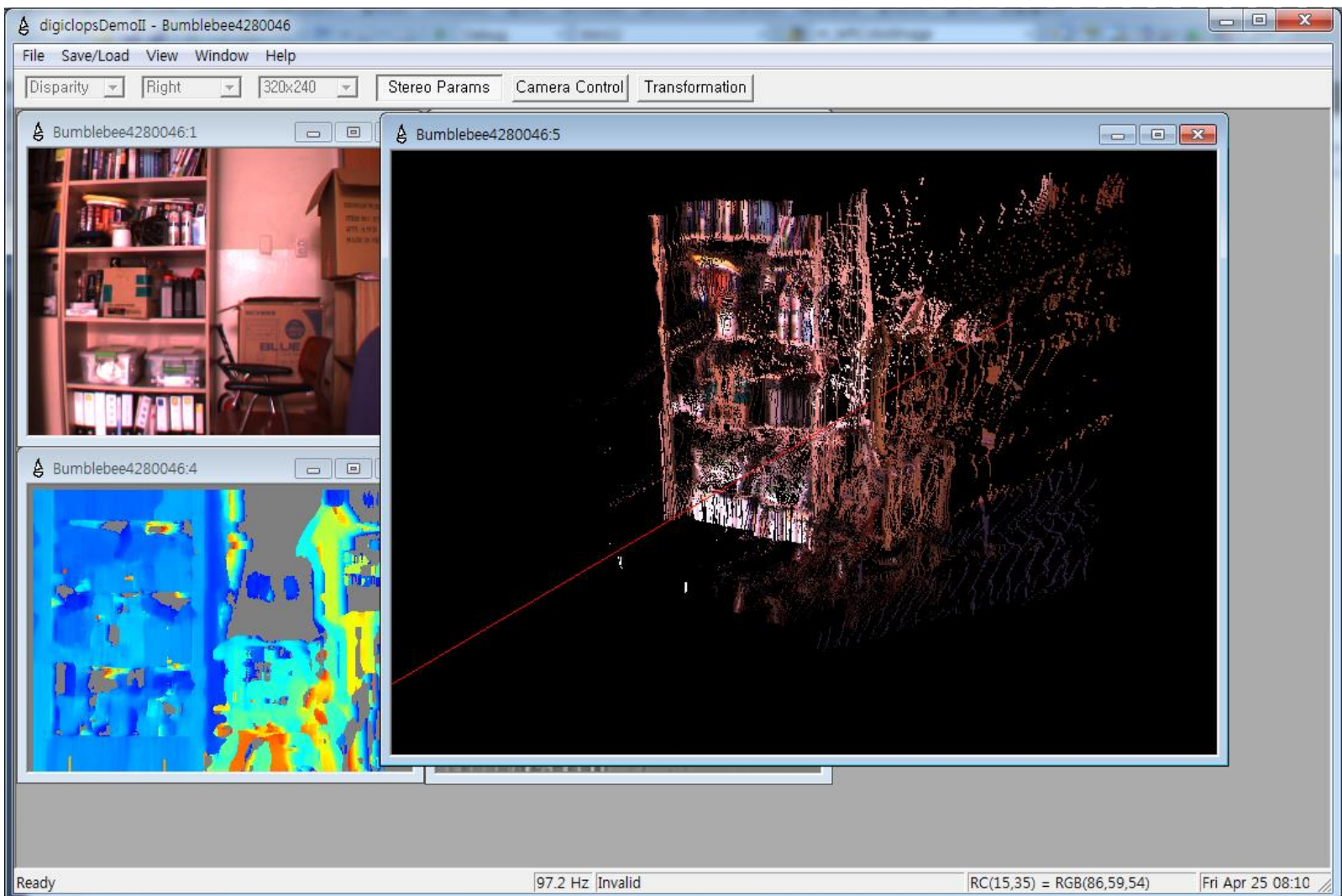
Disparity Validation – Back-forth



Final Output



Start from...



After validation

