



# Metody zpracování dat ze senzorů

Obraz z kamery a hloubkoměru

Pavel Krsek, Libor Wagner

7 a 8. 3. 2018

- ◆ Matematický popis transformací
- ◆ Pořízení hloubkových dat a jejich reprezentace
- ◆ Nelezení rovin - RANSAC
- ◆ Obraz z kamery a detekce hran
- ◆ Vyhledání přímk v obraze - Houghova transformace
- ◆ Matematický model kamery

1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

29 30

31 32

33 34

35 36

37



# Matematické vztahy

bod, vektor, rovina, transformace,  
homogenní souřadnice

1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

29 30

31 32

33 34

35 36

37



## Eukleidovský prostor (kartézský prostor) $\mathbb{R}^2, \mathbb{R}^3$ , ortonormální báze

$\mathbb{R}^2$

$\mathbb{R}^3$

### Bod v prostoru

$$\mathbf{p} = [x, y]$$

$$\mathbf{p} = [x, y, z]$$

### Lineární transformace

$$\begin{aligned} x' &= ax + by + c \\ y' &= dx + ey + f \end{aligned}$$

$$\begin{aligned} x' &= ax + by + cz + d \\ y' &= ex + fy + gz + h \\ z' &= ix + jy + kz + l \end{aligned}$$

### Maticový zápis

$$\mathbf{p}' = \mathbf{R}\mathbf{p} + \mathbf{t}$$

$$\mathbf{p} = [x, y]^T, \mathbf{p}' = [x', y']^T$$

$$\mathbf{R} = \begin{bmatrix} a & b \\ d & e \end{bmatrix}, \mathbf{t} = [c, f]^T$$

$$\mathbf{p}' = \mathbf{R}\mathbf{p} + \mathbf{t}$$

$$\mathbf{p} = [x, y, z]^T, \mathbf{p}' = [x', y', z']^T$$

$$\mathbf{R} = \begin{bmatrix} a & b & c \\ e & f & g \\ i & j & k \end{bmatrix}, \mathbf{t} = [d, h, l]^T$$

$\mathbf{R}$  se nazývá transformační maticí a  $\mathbf{t}$  je translační vektor.

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



Máme dvojici vektorů v  $\mathbb{R}^3$ :  $\mathbf{p} = [p_1, p_2, p_3]$ ,  $\mathbf{q} = [q_1, q_2, q_3]$

## Vzájemný úhel dvou vektorů - Skalární součin

Skalární součin:  $\mathbf{p} \cdot \mathbf{q} = p_1q_1 + p_2q_2 + p_3q_3 = \|\mathbf{p}\| \|\mathbf{q}\| \cos \alpha$

Úhel vektorů:

$$\cos \alpha = \frac{\mathbf{p} \cdot \mathbf{q}}{\|\mathbf{p}\| \|\mathbf{q}\|}$$

## Vektor kolmý na dvojici vektorů - Vektorový součin

Vektorový součin:

$$\mathbf{p} \times \mathbf{q} = \mathbf{n} \|\mathbf{p}\| \|\mathbf{q}\| \sin \alpha$$

$$\mathbf{p} \times \mathbf{q} = \begin{bmatrix} p_2q_3 - p_3q_2 \\ p_3q_1 - p_1q_3 \\ p_1q_2 - p_2q_3 \end{bmatrix} = \begin{bmatrix} 0 & -p_3 & p_1 \\ p_3 & 0 & -p_2 \\ -p_1 & p_2 & 0 \end{bmatrix} \mathbf{q}$$

Normálový vektor:

$$\mathbf{n} = \frac{\mathbf{p} \times \mathbf{q}}{\|\mathbf{p}\| \|\mathbf{q}\| \sin \alpha}, \quad \|\mathbf{n}\| = 1$$

1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

29 30

31 32

33 34

35 36

37



**Definice:** Uspořádanou n-tici čísel  $[h_1, h_2, \dots, h_n]$  nazýváme homogenními souřadnicemi bodu  $\mathbf{p} = [p_1, p_2, \dots, p_{n-1}]$  v  $\mathbb{R}^{n-1}$  pokud platí že  $h_1/h_n = p_1$ ,  $h_2/h_n = p_2$ , ...,  $h_{n-1}/h_n = p_{n-1}$ , kde  $p_1$  až  $p_{n-1}$  jsou kartézské souřadnice bodu  $\mathbf{p}$ .

## Vlastnosti homogenních souřadnic

- ◆ Rozšíření prostoru  $\mathbb{R}^2 \Rightarrow \mathbb{R}^3$  resp.  $\mathbb{R}^3 \Rightarrow \mathbb{R}^4$
- ◆ Popis bodu není jednoznačný:  $\mathbf{p}_H = [h_1, h_2, \dots, h_n] \sim \lambda[h_1, h_2, \dots, h_n]$ ,  $\lambda \neq 0$
- ◆ Převod z/do homogenních souřadnic

$$\begin{aligned} \mathbb{R}^2 \\ \mathbf{p} = [u, v] &\Rightarrow \mathbf{p}_H = [u, v, 1] \\ \mathbf{p}_H = [u, v, w] &\Rightarrow \mathbf{p} = \left[\frac{u}{w}, \frac{v}{w}\right] \end{aligned}$$

$$\begin{aligned} \mathbb{R}^3 \\ \mathbf{p} = [x, y, z] &\Rightarrow \mathbf{p}_H = [x, y, z, 1] \\ \mathbf{p}_H = [x, y, z, w] &\Rightarrow \mathbf{p} = \left[\frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right] \end{aligned}$$

- ◆ Lineární transformace souřadnic

$$\mathbb{R}^2 : \quad \mathbf{p}'_H = \mathbf{P}^{3 \times 3} \mathbf{p}_H \quad \mathbb{R}^3 : \quad \mathbf{p}'_H = \mathbf{P}^{4 \times 4} \mathbf{p}_H$$

- ◆ Afinní transformace souřadnic

$$\mathbf{p}'_H = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{p}_H$$

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



$\mathbb{R}^2$  - přímka

$\mathbb{R}^3$  - rovina

## Parametrické vyjádření (parametry $s, t$ )

$$u = u_0 + a_1s$$

$$v = v_0 + a_2s$$

$$\mathbf{p} = \mathbf{p}_0 + \mathbf{a}s$$

$$x = x_0 + a_1s + b_1t$$

$$y = y_0 + a_2s + b_2t$$

$$z = y_0 + a_3s + b_3t$$

$$\mathbf{p} = \mathbf{p}_0 + \mathbf{a}s + \mathbf{b}r$$

## Obecná rovnice (maticově v homogenních souřadnicích)

$$au + bv + c = 0$$

$$\mathbf{A}\mathbf{p}_H = \mathbf{0}, \text{ kde } \mathbf{A} = [a, b, c]$$

$$ax + by + cz + d = 0$$

$$\mathbf{A}\mathbf{p}_H = \mathbf{0}, \text{ kde } \mathbf{A} = [a, b, c, d]$$

## Normála $\mathbf{n}$

$$\mathbf{n} = [a, b] / \sqrt{a^2 + b^2}$$

$$\mathbf{n} = [a, b, c] / \sqrt{a^2 + b^2 + c^2}$$

## Vzdálenost $\Delta$ bodu $\mathbf{p}_1$

$$\Delta = \frac{|ax_1 + by_1 + c|}{\sqrt{a^2 + b^2}}$$

$$\Delta = \frac{|ax_1 + by_1 + cz_1 + d|}{\sqrt{a^2 + b^2 + c^2}}$$

1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

29 30

31 32

33 34

35 36

37



### Parametrické vyjádření (parametr $s$ )

$$\begin{aligned} x &= x_0 + a_1s \\ y &= y_0 + a_2s \quad \equiv \quad \mathbf{p} = \mathbf{p}_0 + \mathbf{a}s \\ z &= z_0 + a_3s \end{aligned}$$

### Průsečík dvou rovin (řešení soustavy lineárních rovnic)

$$\begin{aligned} a_1x + b_1y + c_1z + d_1 &= 0 \\ a_2x + b_2y + c_2z + d_2 &= 0 \end{aligned}$$

### Řešení soustavy homogenních lineárních rovnic (SVD)

- ◆ Soustava rovnic:  $\mathbf{A}\mathbf{x} = \mathbf{0}$ , matice  $\mathbf{A}$  je známá, hledáme řešení  $\mathbf{x} \neq \mathbf{0}$ .
- ◆ SVD dekompozice:  $\mathbf{A} = [\mathbf{U}|\mathbf{U}_0]_{m \times m} \begin{bmatrix} \mathbf{S}_{n \times n} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} [\mathbf{V}|\mathbf{V}_0]_{n \times n}^T$
- ◆ Vektor  $\mathbf{V}_0$  je pravým singulárním vektorem odpovídající nulovému vlastnímu číslu matice  $\mathbf{A}$ . Řešením je  $\mathbf{x} = \mathbf{V}_0$ .
- ◆ Pokud matice  $\mathbf{A}$  nemá nulové vlastní číslo a  $\mathbf{V}_0$  odpovídá nejmenšímu z nich pak řešení  $\mathbf{x} = \mathbf{V}_0$  minimalizuje normu  $\mathbf{A}\mathbf{x}$ .

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



# 3D data

mrak bodů, hloubková mapa  
hloubkové snímače (rangefinders)

1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

29 30

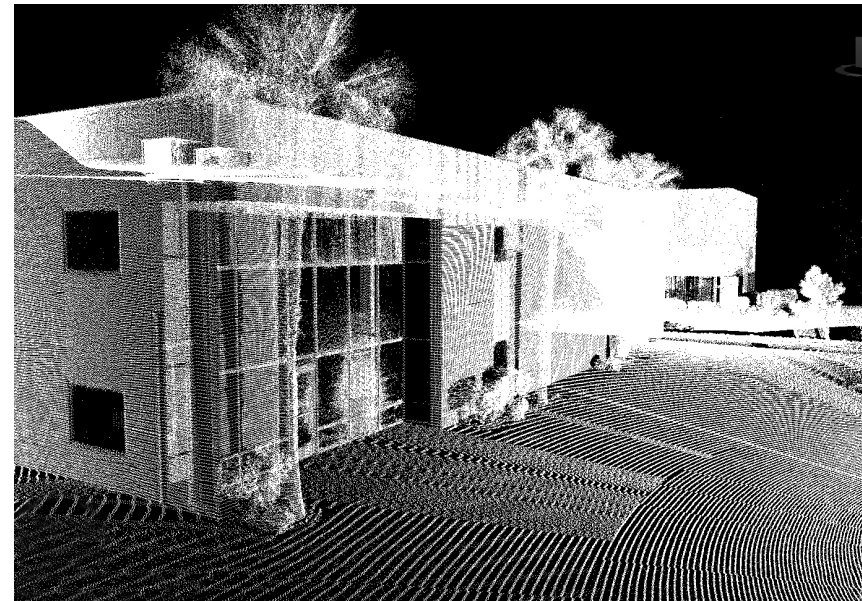
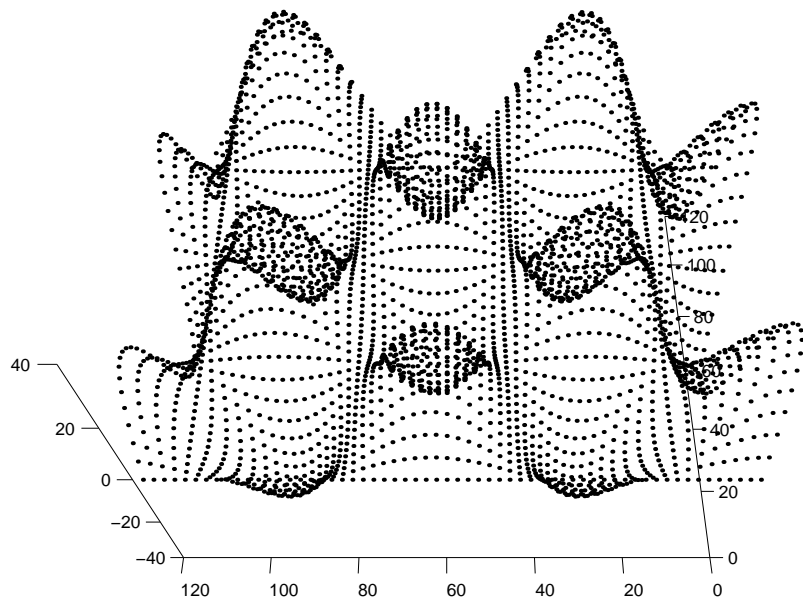
31 32

33 34

35 36

37





**Definice:** Množina bodů  $P = p_i : i = 1...N$

- ◆ Datová struktura generovaná většinou hloubkoměrů
- ◆ Body bez definovaného vzájemného vztahu
- ◆ Body mohou tvořit plochu či těleso

Orázky: <http://www.revittunes.com> a autor

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



Depth map

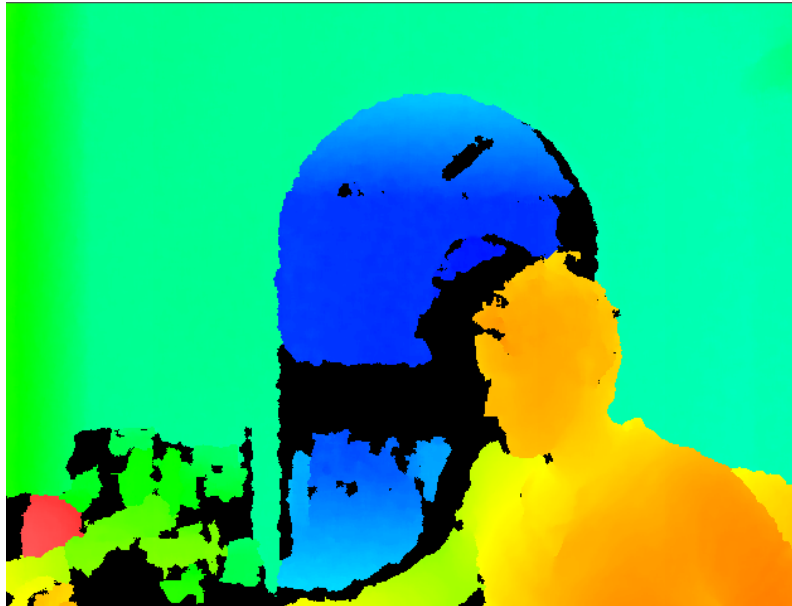


RGB image

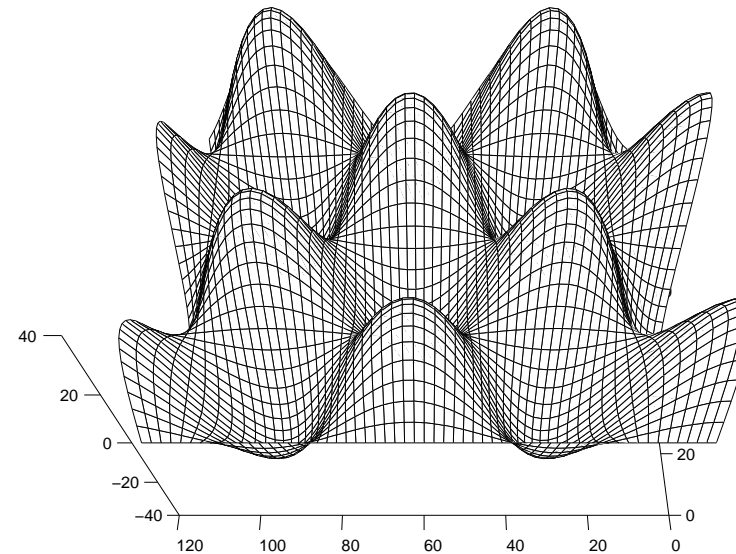
**Definice:** 2D matice čísel  $M_{i,j} \in \mathbb{R}$  která reprezentuje 3. souřadnici ( $z$ ) zobrazených bodů. 1. a 2. souřadnice ( $x, y$ ) je definována polohu v této matici.

- ◆ Jednoduchá topologická struktura
- ◆ Rychlé zpracování v matici (filtrování).
- ◆ Hloubková mapa je obvykle v souřadné soustavě senzoru

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



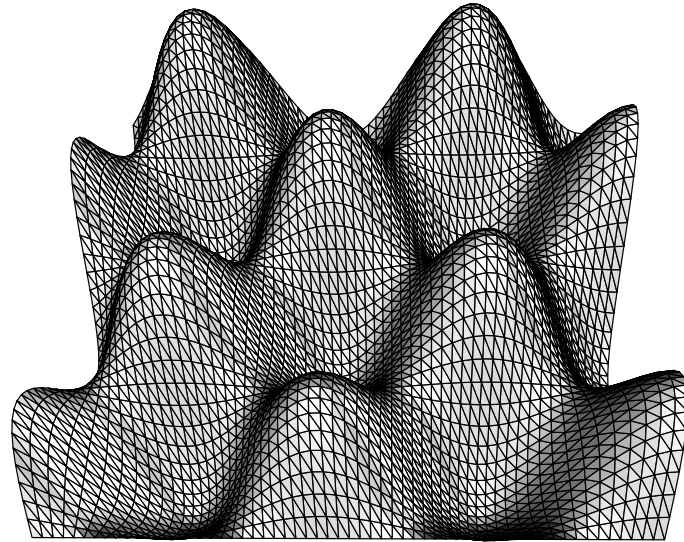
Hloubková mapa



“Mesh”

- ◆ Je nutno řešit reprezentace nezměřených bodů (NaN)
- ◆ Není zřejmé kdy je zobrazená plocha nespojitá
- ◆ Hloubková mapa může popsat pouze funkci  $z = f(x, y)$

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



**Definice:** Množina bodů  $P = \{p_i : i = 1 \dots N\}$  a množina trojúhelníků  $T = \{t_m : m = 1 \dots M\}$ , kde  $t_m = \{[p_i, p_j, p_k] : i \neq j \neq k\}$  je trojice bodů z  $P$ .

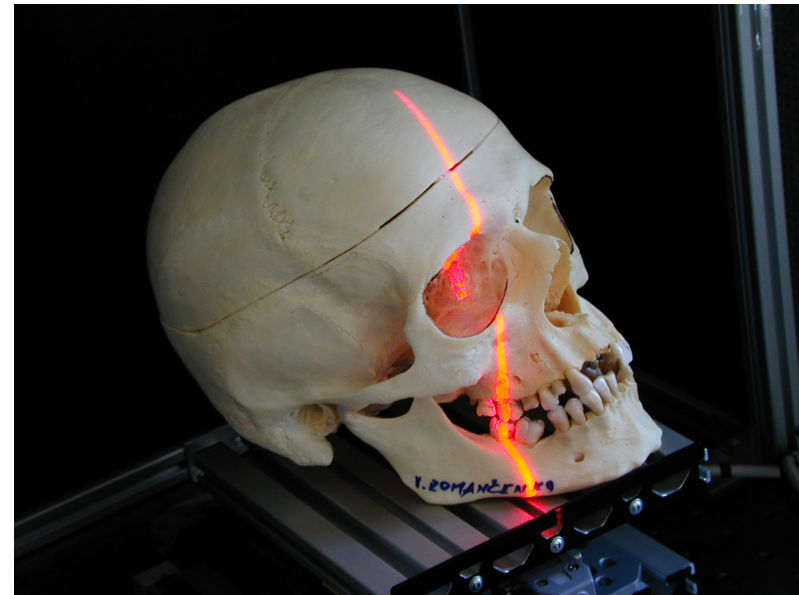
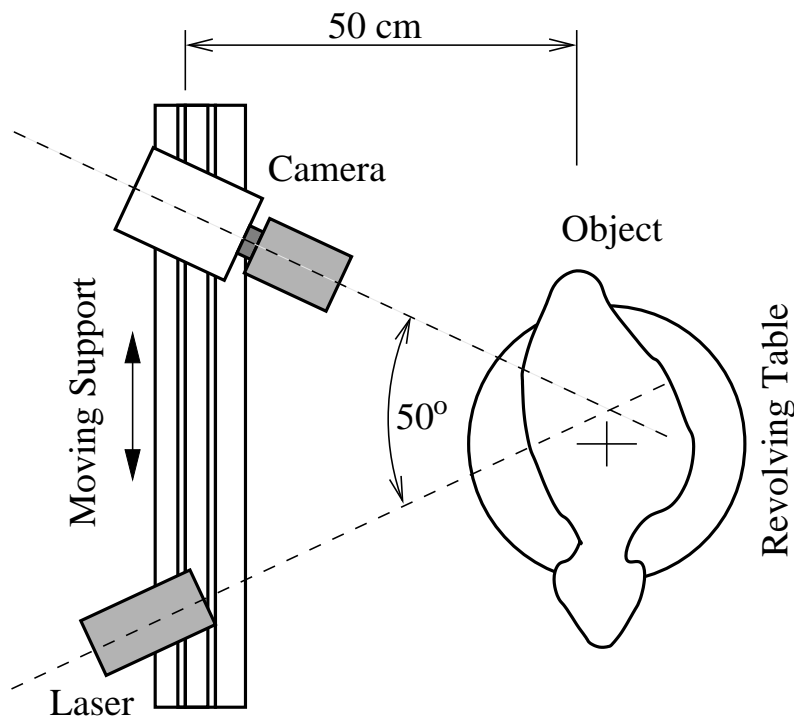
- ◆ Body s relací “sousedství”
- ◆ Trojúhelníky definují povrch (roviny)
- ◆ Trojúhelník může být doplněn o směr normály (vnitřní/vnější prostor)

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



## Hloubkoměr se strukturovaným světlem

- ◆ Projektor (laser) promítá známý vzor
- ◆ Kamera snímá vzor promítnutý na povrch objektu
- ◆ Informace o hloubce se získá triangulací
- ◆ Výstupem je obvykle hloubková mapa



1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

29 30

31 32

33 34

35 36

37

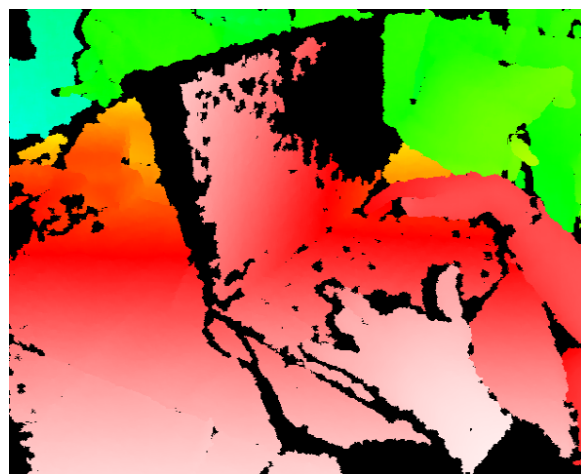


## Základní vlastnosti hloubkového snímače

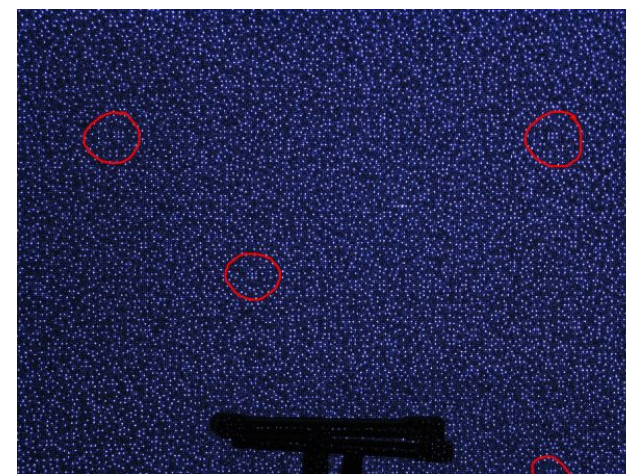
- ◆ Hloubkoměr založený na strukturovaném světle
- ◆ IR vzor promítaný na objekt (matice 3x3)
- ◆ IR kamera měří hloubku, RGB kamera snímá texturu
- ◆ Rozlišení hloubkoměru 640x480, Rozlišení RGB kamery 1280x1024



Obrázek z IR kamery



Hloubková mapa



Promítaný vzor (IR)

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



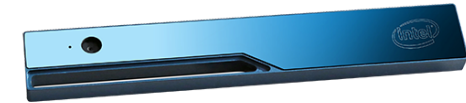
## Parametry hloubkových snímačů



Kinect Xbox 360



Orbbec Astra



Realsense R200

FOV [stupňů]:	57 x 45	60 x 49.5	59x 45.5
Rozsah [m]:	1.5 ... 3.5	0.6 ... 8.0	0.5 ... 3.5 (4.0)
Chyba XY [mm]:	10 (2.5m)	7.2 (3m)	—
Chyba Z [mm]:	10 (2.5m)	12.7 (3m)	10 (2m)
Rozlišení [px]:	640x480	640x480	640x480

TurtleBot je osazen snímačem Realsense R200, nebo Orbbec Astra.

## Výrobci

- ◆ <https://developer.microsoft.com/en-us/windows/kinect>
- ◆ <https://software.intel.com/en-us/realsense>
- ◆ <https://orbbec3d.com>



# Metoda RANSAC

detekce a segmentace rovin

1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

29 30

31 32

33 34

35 36

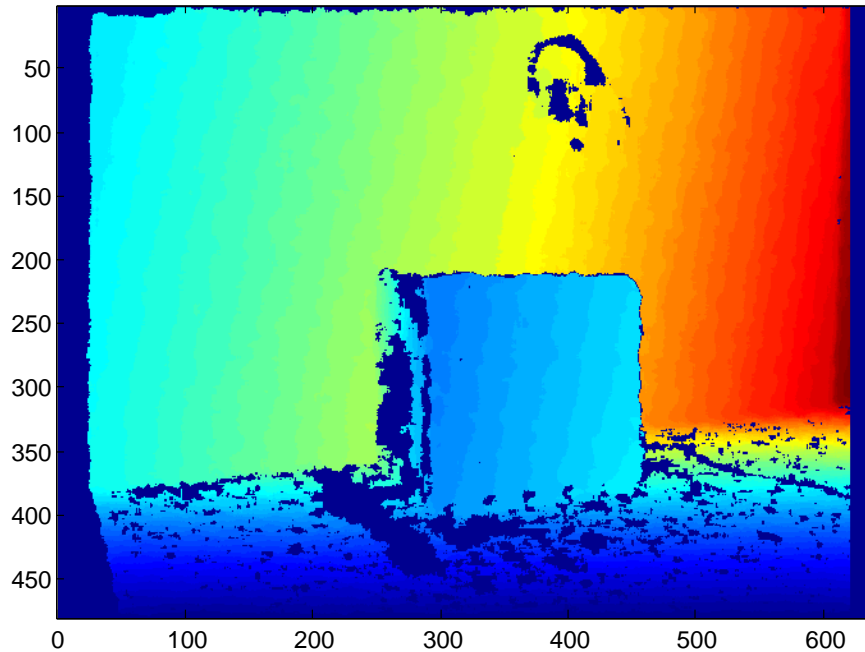
37





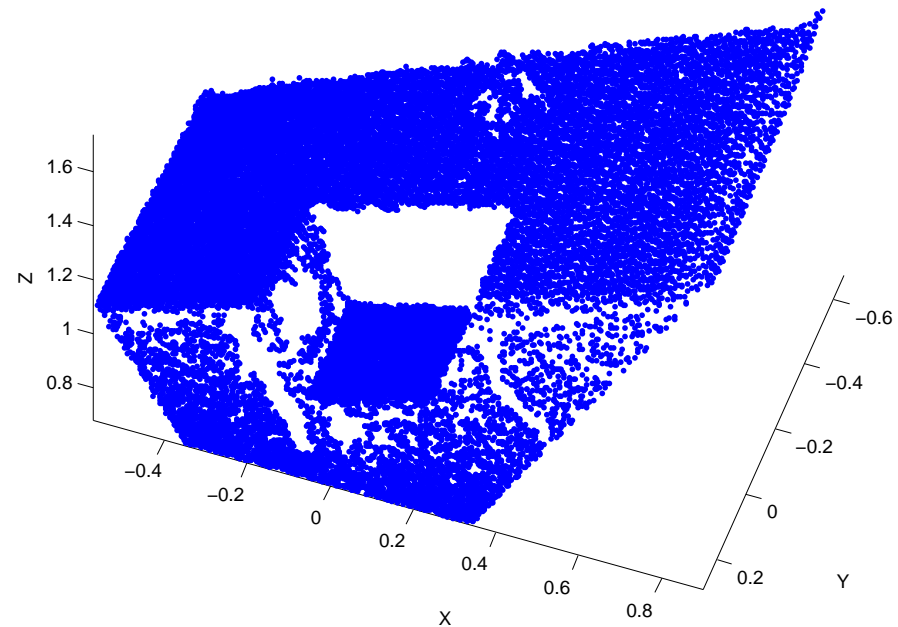
Vstupem pro detekci rovin je mrak bodů

Hloubková mapa



`d = turtle.get_depth_image()`  
`element = d[řádek, slupek] ≡ hloubce`

Mrak bodů



`x = turtle.get_point_cloud()`  
`element = x[řádek, slupek, vrstva]`  
`vrstava ∈ [0, 1, 2] ≡ x, y, z`

1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

29 30

31 32

33 34

35 36

37



RANSAC = Random sample consensus for model fitting

## Základní vlastnosti algoritmu

- ◆ Algoritmus RANSAC slouží pro odhadování parametrů matematického modelu na základě naměřených dat.
- ◆ Metoda je založena na náhodném vzorkování dat, kdy vybírám minimální množství vzorků, které je nutné pro určení parametrů modelu.
- ◆ Je možné hledat i model, který odpovídá relativně malému počtu dat.
- ◆ Metoda je málo citlivá na přítomnost odlehlých hodnot (outlier) v datech.
- ◆ Obecně není zaručeno, že určíme správné parametry modelu.

1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

29 30

31 32

33 34

35 36

37



1. Máme  $n$  bodů  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ , která chceme aproximovat modelem definovaným  $m$  body ( $m \leq n$ , rovina  $m = 3$ ).
2. Nastavíme čítač iterací na  $k = 1$ .
3. Náhodně vybereme  $m$  bodů z množiny  $\mathbf{X}$  a vypočteme model (určíme parametry).
4. Určíme počet bodů  $t$  (support) z množiny  $\mathbf{X}$ , které odpovídají modelu s uvažování zvolené tolerance  $e$ .
5. Zpřesníme model za pomoci všech bodů, které mu odpovídají a aktualizujeme počet těchto bodů pro zpřesněný model  $t$ .
6. Uložíme model pokud je počet bodů  $t$  větší než u dosud uloženého modelu.
7. Inkrementujeme čítač iterací  $k = k + 1$ . Pokud čítač  $k$  nedosáhl maximálního zadaného počtu opakování  $K$  opakujeme algoritmus od bodu 3. Jinak považujeme za výsledek model s maximálním počtem odpovídajících bodů  $t$ .
8. Zpřesníme model za pomoci všech bodů, které mu odpovídají.

1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

29 30

31 32

33 34

35 36

37



## Počet iterací

- ◆ Nechť  $p$  je pravděpodobnost, že algoritmus vybere v některé z iterací pouze body odpovídající modelu (inliers).
- ◆ Na základě znalosti dat můžeme odhadnout pravděpodobnost, že jedním pokusem vybereme bod odpovídající modelu (inliers) jako  $w = \# \text{ inliers} / \# \text{ outliers}$ .
- ◆ Potom pravděpodobnost, že algoritmus v žádné z  $k$  iterací nevybere pouze modelu odpovídající body je

$$1 - p = (1 - w^m)^k,$$

kde  $m$  je počet vybíraných bodů a  $w^m$  pravděpodobnost, že budou vybrány v daném kroku pouze body odpovídající modelu.

- ◆ Na základě požadované pravděpodobnosti nalezení správného modelu můžeme stanovit počet iterací

$$k = \frac{\log(1 - p)}{\log(1 - w^m)}$$

## Alternativní ukončení algoritmu

- ◆ Pokud víme, kolik bodů (minimálně) by mělo odpovídat modelu, můžeme stanovit práh  $T$  na počet bodů odpovídajících modelu  $t$ .
- ◆ Pokud počet odpovídajících bodů přesáhne tento práh  $T$  ukončíme iterace.

1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

29 30

31 32

33 34

35 36

37



## Vstup algoritmu

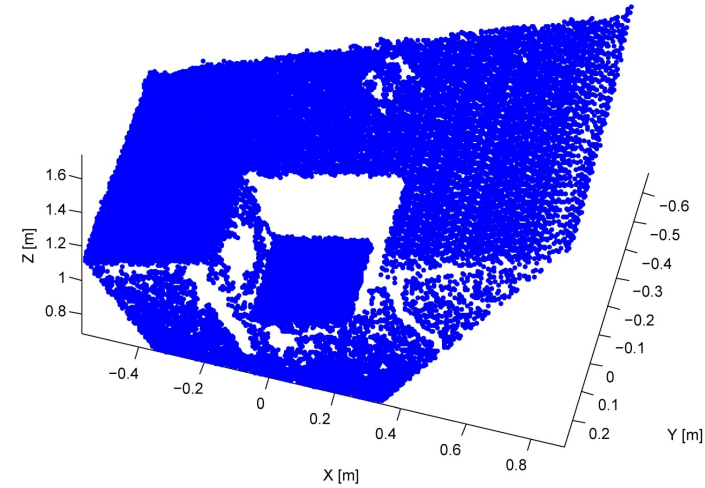
- ◆ Použitý příklad má 256506 bodů
- ◆ V datech je více než jedna rovina

## Rozšíření algoritmu pro více rovin

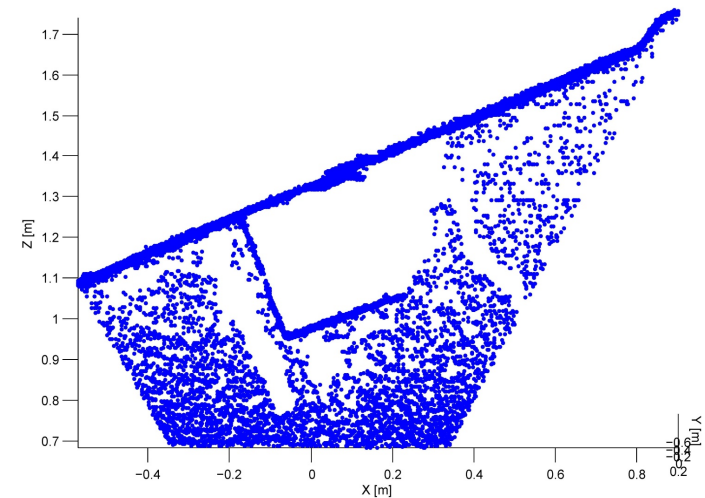
1. Algoritmem RANSAC vyhledáme v datech rovinu.
2. Z dat odebereme body které odpovídají modelu této roviny.
3. Algoritmus opakujeme od 1 s redukovanými daty.

## Zastavení algoritmu

- ◆ Po nalezení zadaného počtu rovin
- ◆ Pokud zbývá méně než zvolený počet nezařazených bodů
- ◆ Pokud nacházené roviny nerepresentují dostatečný počet bodů



data z hloubkoměru

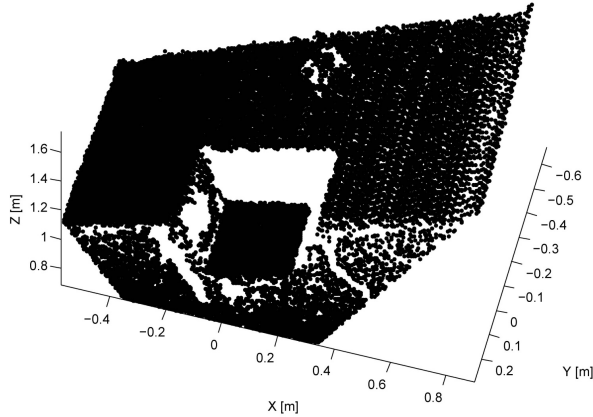


data v půdorysu

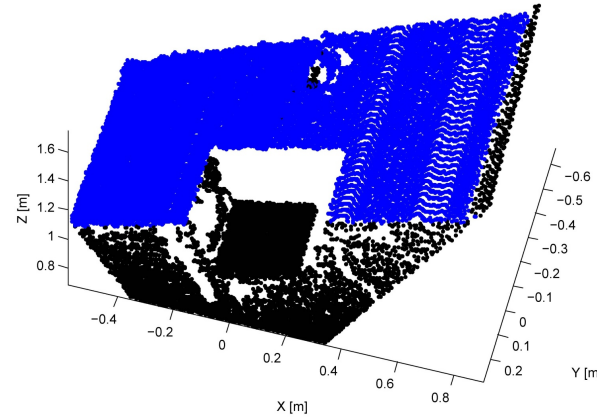
1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



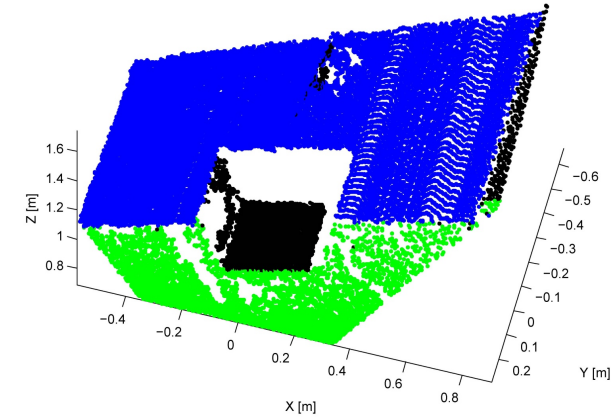
## Výsledek opakovaného algoritmu RANSAC



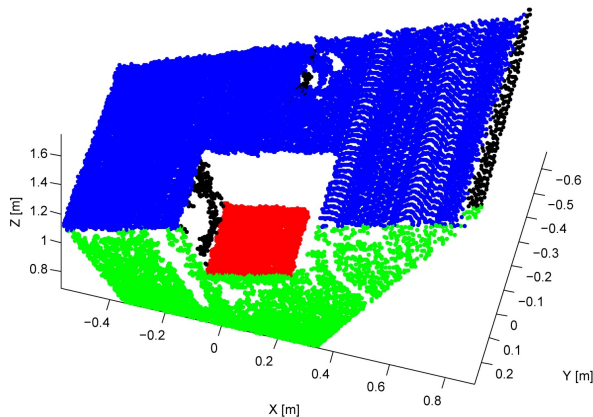
Vstupní mrak bodů



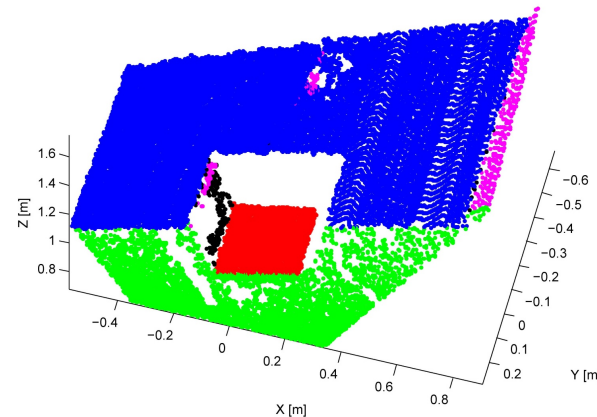
Detekována 1. rovina



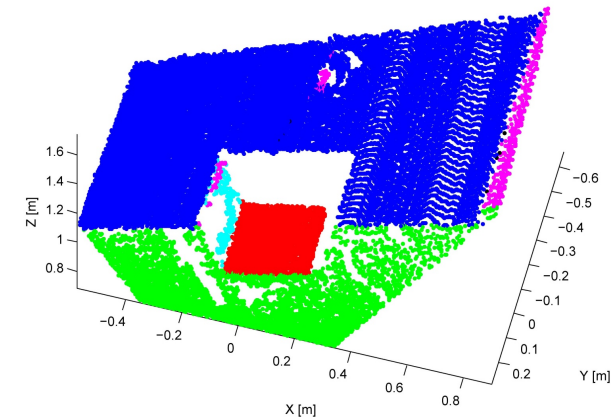
Detekována 2. rovina



Detekována 3. rovina



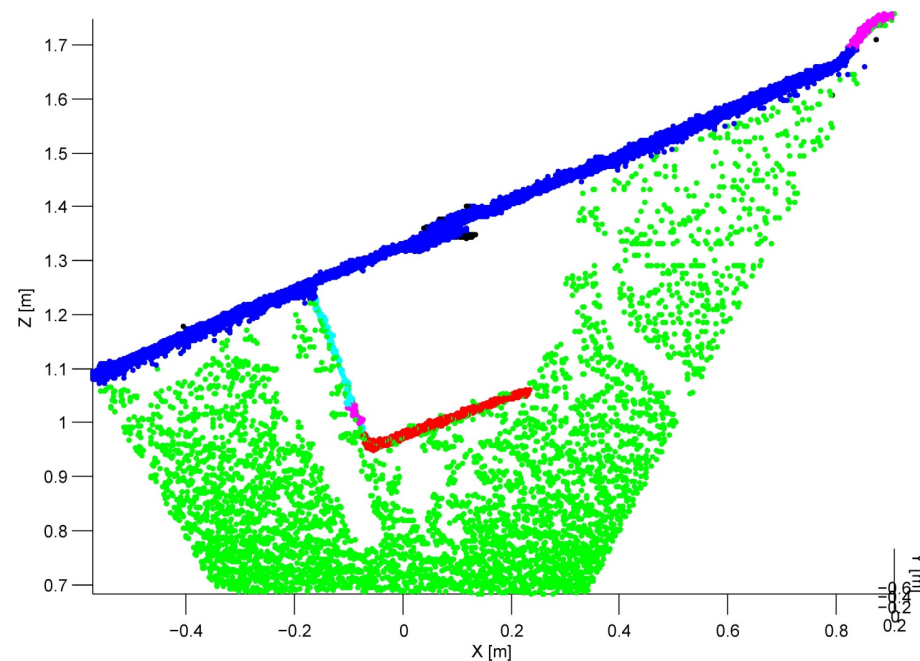
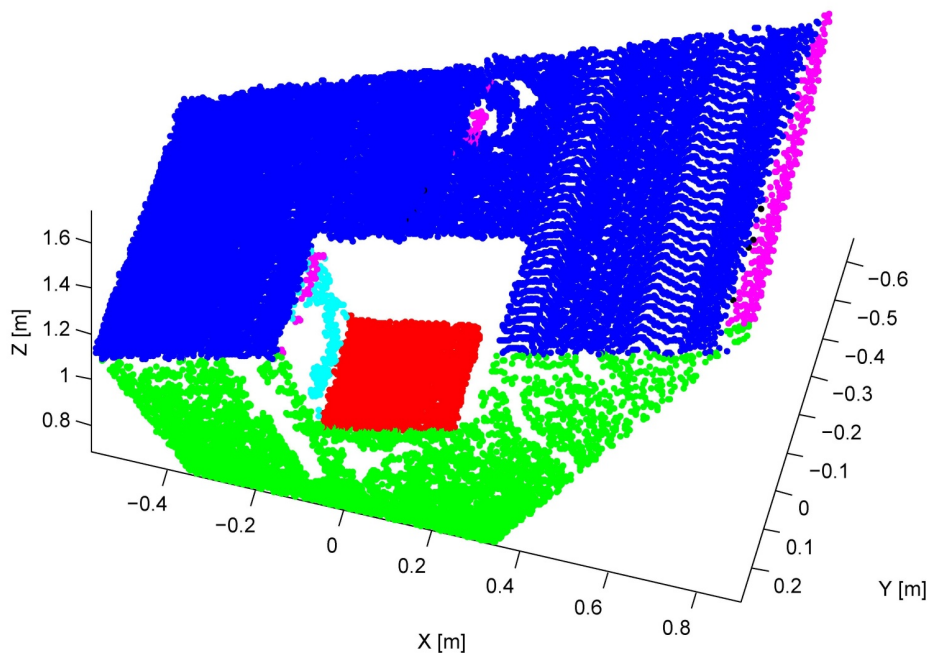
Detekována 4. rovina



Detekována 5. rovina

- ◆ Algoritmus končí pokud zbývá méně jak 1000 nezařazených bodů.
- ◆ Detekováno bylo celkem 5 rovin.

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



## Parametrický popis nalezených rovin:

Rovina	a	b	c	d
1 modrá	0.384	0.068	-0.921	1.196
2 zelená	-0.005	-0.998	-0.060	0.329
3 červená	0.336	0.061	-0.940	0.896
4 fialová	0.466	0.068	-0.882	1.104
5 sv. modrá	0.941	-0.013	0.338	-0.255

## Vzájemné úhly rovin [°]:

	1	2	3	4	5
1	0.0	90.9	2.9	5.2	87.2
2	90.9	0.0	90.3	91.0	90.7
3	2.9	90.3	0.0	8.1	90.1
4	5.2	91.0	8.1	0.0	82.0
5	87.2	90.7	90.1	82.0	0.0

## Úhel normál rovin ke svislici [°]:

	1	2	3	4	5
	86.1	3.4	86.5	86.1	89.2

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



# Práce s obrazem z kamery

Houghova transformace - detekce přímek

1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

29 30

31 32

33 34

35 36

37





## RGB snímek



```
I = turtle.get_rgb_image()
```

```
element = I [řádek, sloupec, barva] ∈ ⟨0; 255⟩
```

```
barva ∈ [0, 1, 2] ≡ B, G, R
```

1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

29 30

31 32

33 34

35 36

37

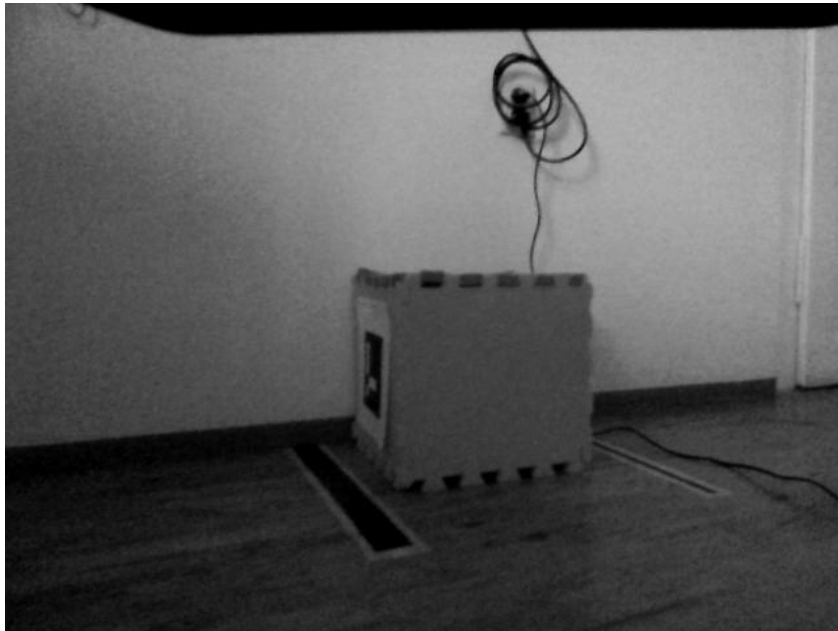


RGB snímek

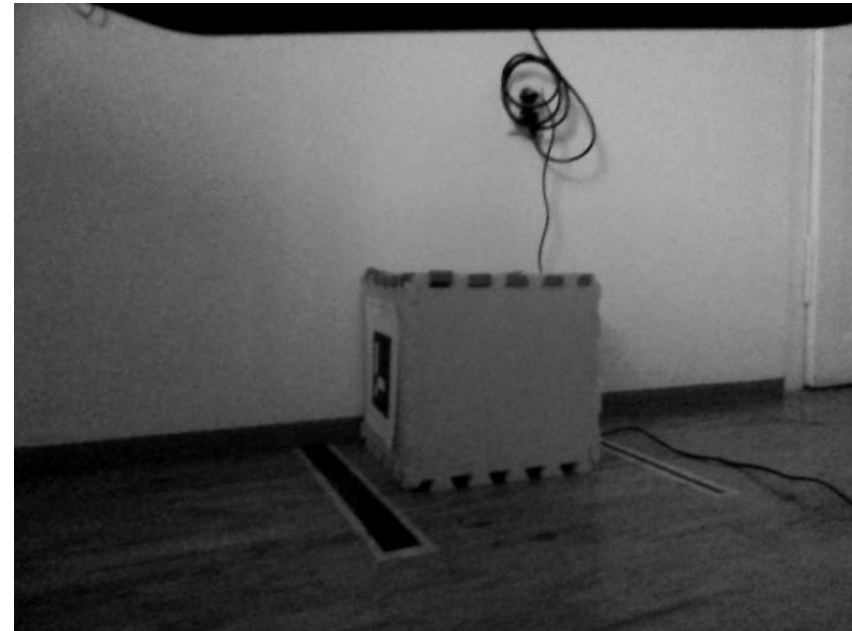
## Způsob převodu

1.  $I = 0.333I_R + 0.333I_G + 0.333I_B$

2.  $I = 0.299I_R + 0.587I_G + 0.114I_B$



1



2

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



**Jednoduše:** Hrana je místo ve snímku, kde se výrazně mění jeho jas (intenzita).

## Sobelův hranový detektor

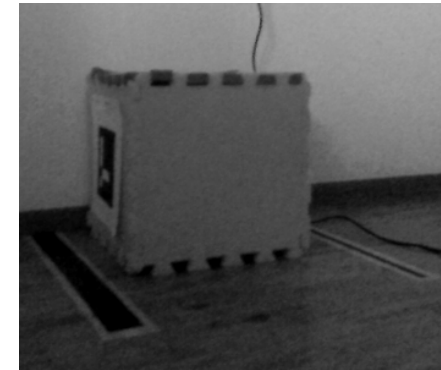
- ◆ Konvoluce obrazu s maskou pro svislé a vodorovné hrany:

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}, G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

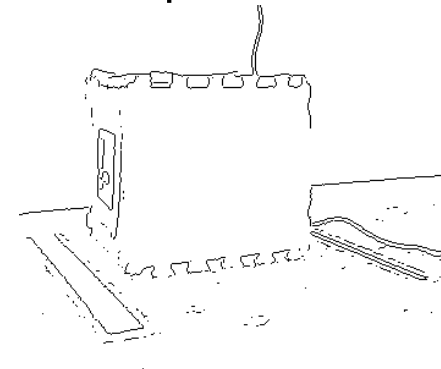
- ◆ Intenzita hrany v daném bodě je  $G = \sqrt{G_x^2 + G_y^2}$ .
- ◆ Nastavuje se práh intenzity  $G$ .

## Metoda Laplacián/Gaussián

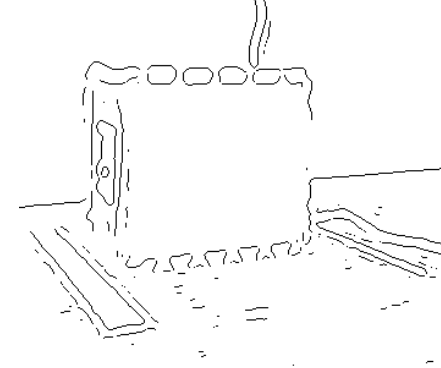
- ◆ Detekce používá Laplaceův operátor kombinovaný s Gausovským filtrem.
- ◆ Detekují se průchody nulou 2. derivace obrazové funkce
- ◆ Nastavuje se práh na detekci hrany a  $\sigma$  filtru.



Vstupní snímek



Sobelův detektor



Laplacián/Gaussián

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



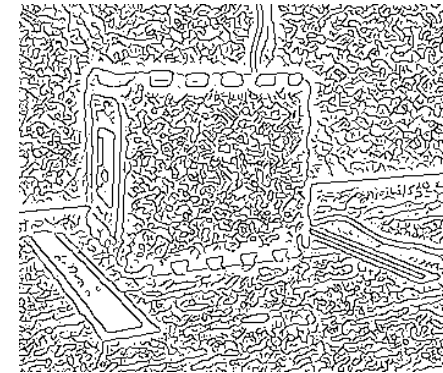
## Popis algoritmu

1. Omezení vlivu šumu Gaussovým filtrem.
2. Hrany jsou detekovány Sobelovým operátorem.
3. Intenzita hrany vypočtena  $G = \sqrt{G_x^2 + G_y^2}$ , směr hrany  $\Theta = \text{atan2}(G_y, G_x)$ .
4. Provedeno "ztenčení" hran ve směru kolmém na hranu (4 směry). Ponechá pouze body s maximální intenzitou hrany v daném směru.
5. Dvojitě prahování a sledování hrany:
  - ◆ Jsou vybrány body, které mají intenzitu větší než horní práh  $T_1$  (silné hrany).
  - ◆ Postupně jsou jako hranové body označovány body sousedící s hranovými, pokud je intenzita hrany v daném bodě větší než dolní práh  $T_2$ .

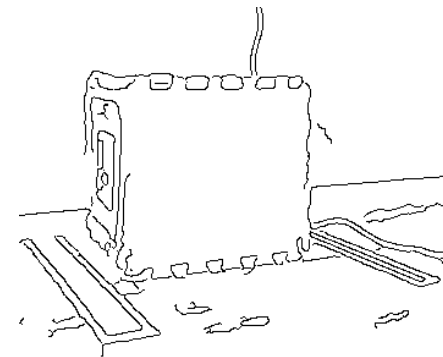
## Nastavování parametrů

- ◆ Dva prahy  $T_1$ ,  $T_2$  a  $\sigma$  Gaussova filtru.

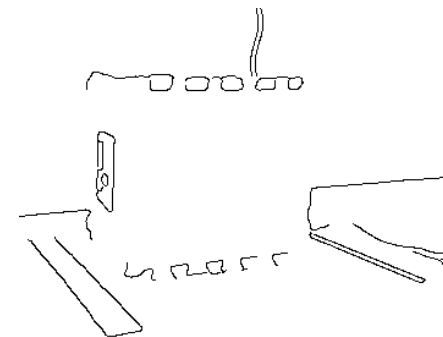
$$T_2 = 0.4T_1$$



$$\sigma = \sqrt{2}, T_1 = 0.01$$



$$\sigma = \sqrt{2}, T_1 = 0.1$$



$$\sigma = \sqrt{2}, T_1 = 0.3$$

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



## Principy Houghovy transformace

- ◆ Hledáme instance křivky definované rovnicí  $f(\mathbf{u}, \mathbf{a}) = 0$ , kde  $\mathbf{u}$  jsou souřadnice ve snímku a  $\mathbf{a}$  je parametrizace hledané křivky.
- ◆ Vstupem jsou obvykle body na hranách (edge detection).
- ◆ Každý hraniční bod může ležet na nekonečně mnoho instancích hledaných křivek.
- ◆ Transformujeme body ze souřadnic snímku do souřadnic parametrů a křivky.

## Algoritmus Houghovy transformace

1. Provedeme kvantizaci vektoru parametrů  $\mathbf{a}$ .
2. Vytvoříme  $n$ -rozměrné pole akumulátoru  $\mathbf{A}(\mathbf{a})$ , které strukturou odpovídá parametrům  $\mathbf{a}$ .
3. Pro každý nalezený hraniční bod  $\mathbf{u} = [u, v]$  v obraze budeme inkrementovat všechny prvky akumulátoru  $\mathbf{A}(\mathbf{a})$  pro něž platí  $f(\mathbf{u}, \mathbf{a}) = 0$ .
4. Po započtení všech hraničních bodů platí, že lokální maxima v akumulátoru  $\mathbf{A}(\mathbf{a})$  odpovídají jednotlivým instancím křivky  $f(\mathbf{u}, \mathbf{a}) = 0$

1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

29 30

31 32

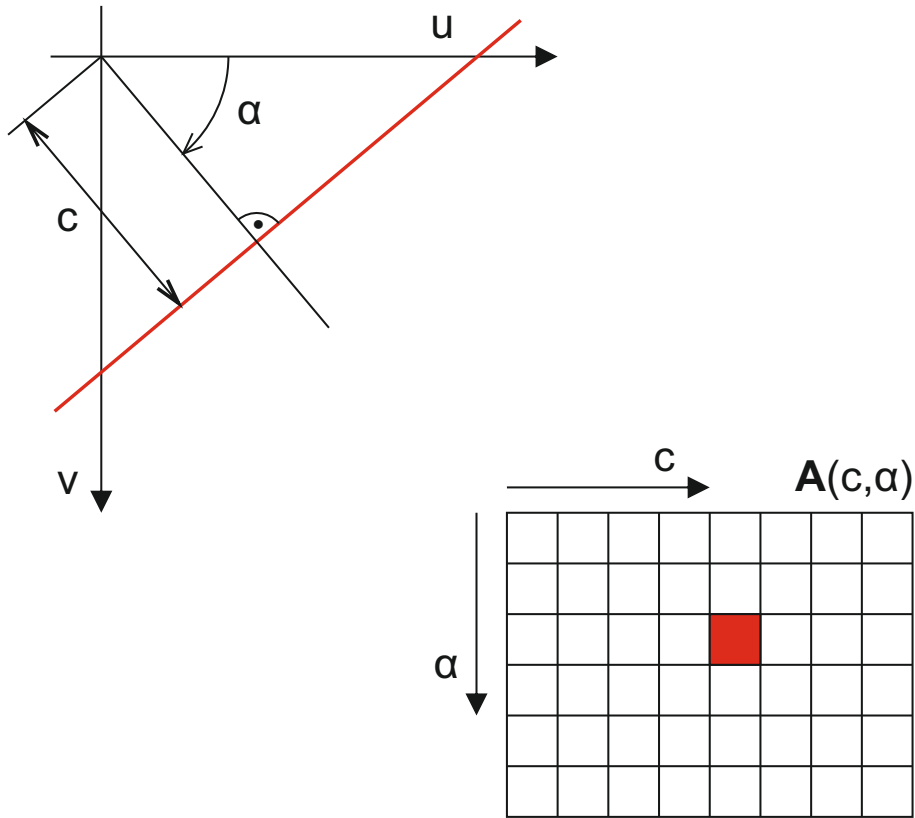
33 34

35 36

37

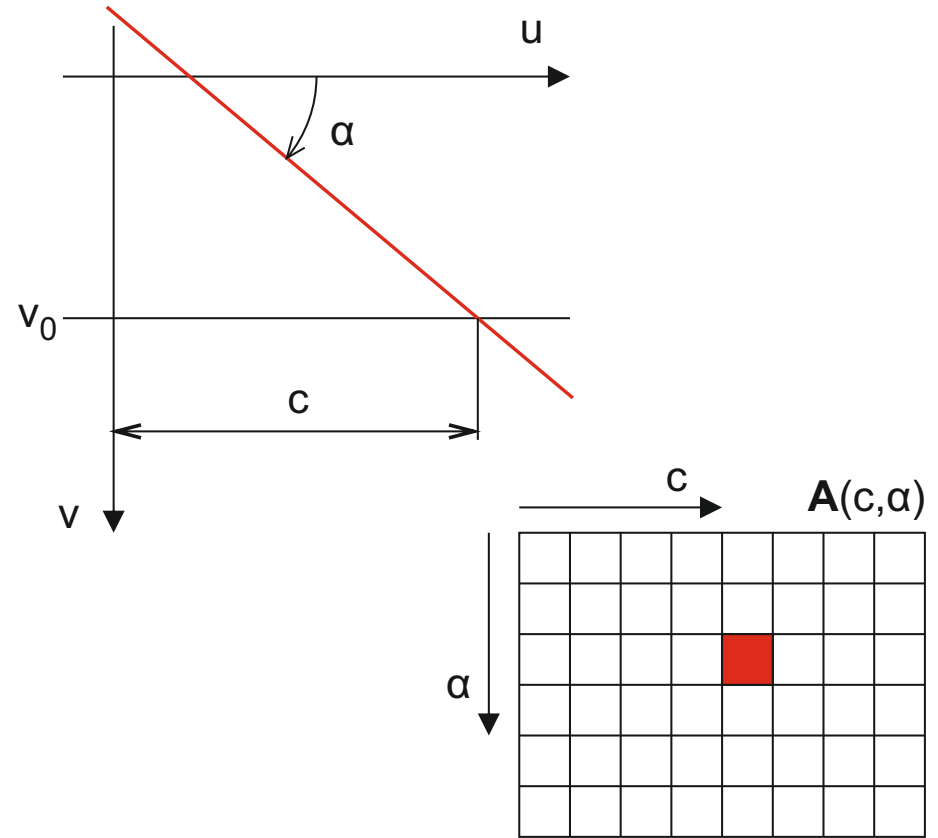


## Obvyklá parametrizace



$$u \cos \alpha + v \sin \alpha = c$$

## Alternativní parametrizace



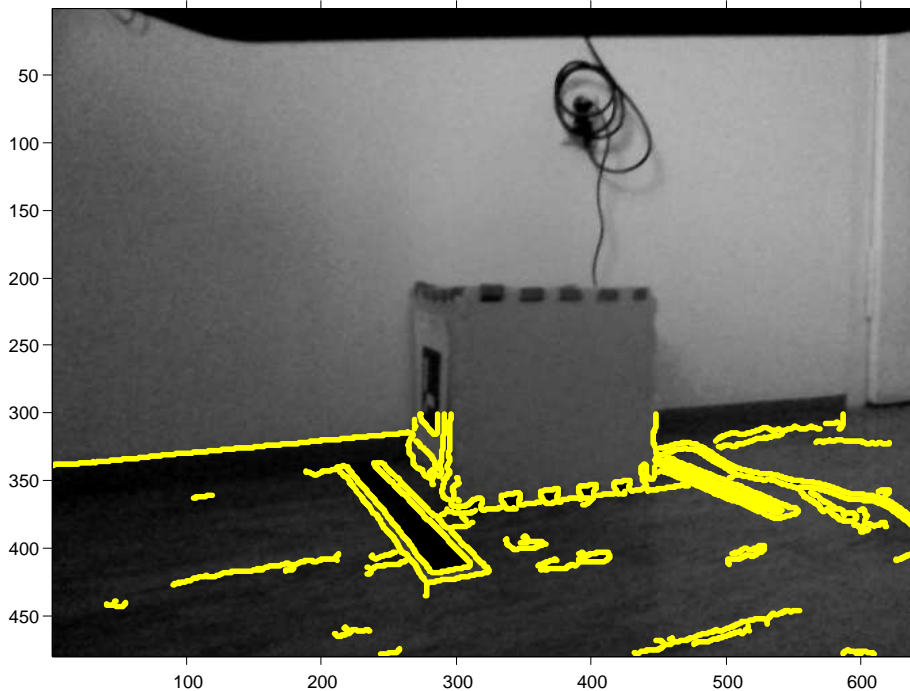
$$u - \frac{1}{\tan \alpha} (v_0 - v) = c$$

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	

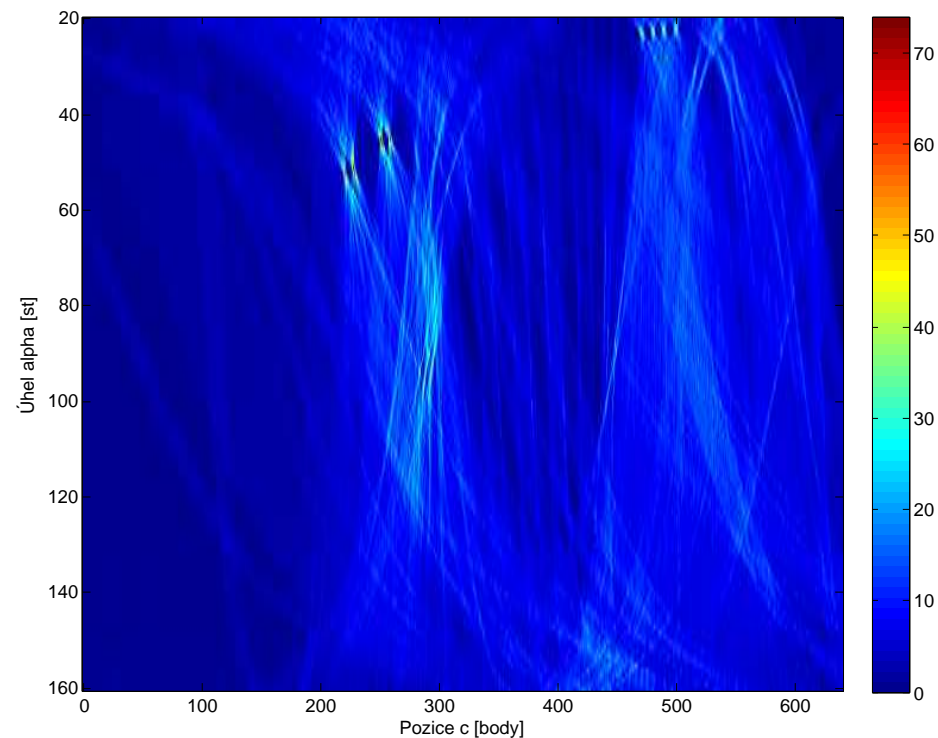


## Použití Houghovy transformace

- ◆ Vstupem je seznam bodů na hranách
- ◆ Hledáme parkovací stání na podlaze, proto se omezíme na hranové body s řádkovou souřadnicí  $> 300$ .
- ◆ Rozsah parametrů volíme  $\alpha \in \langle 20^\circ; 160^\circ \rangle$  a  $c \in \langle 0; 640 \rangle$  obrazových bodů.
- ◆ Kvantizace je  $1^\circ$  a 1 obrazový bod.



Vstupující hranové body



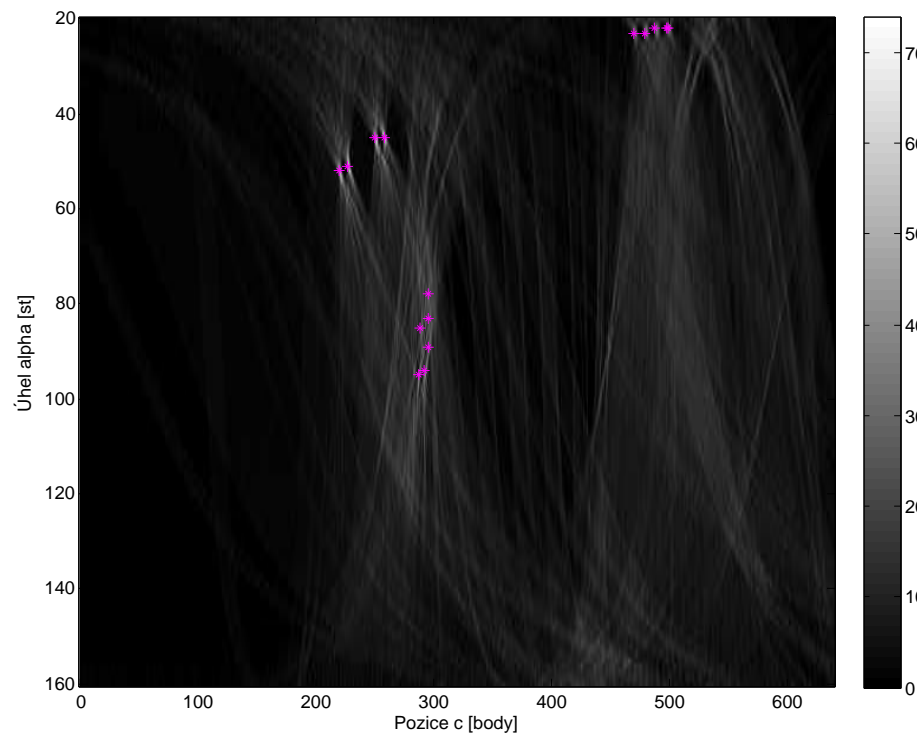
Výsledek transformace

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	

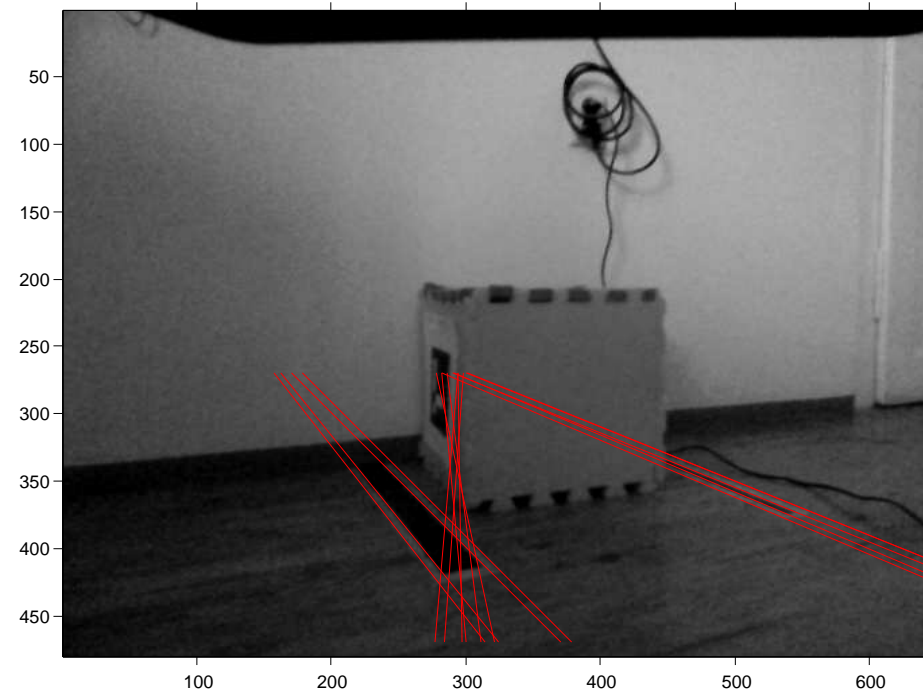


## Lokalizace hran

- ◆ Hledáme maxima v transformované obrázku.
- ◆ Maximu zároveň splňuje požadavek na minimální podporu  $> 30$  hranových bodů.
- ◆ Na základě parametrů  $\alpha$  a  $c$  nalezených maxim konstruueme přímky.



Detekovaná maxima



Odpovídající přímky

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	





# Model kamery

Přepoččet ze souřadné soustavy kamery do  
globální souřadné soustavy

1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

29 30

31 32

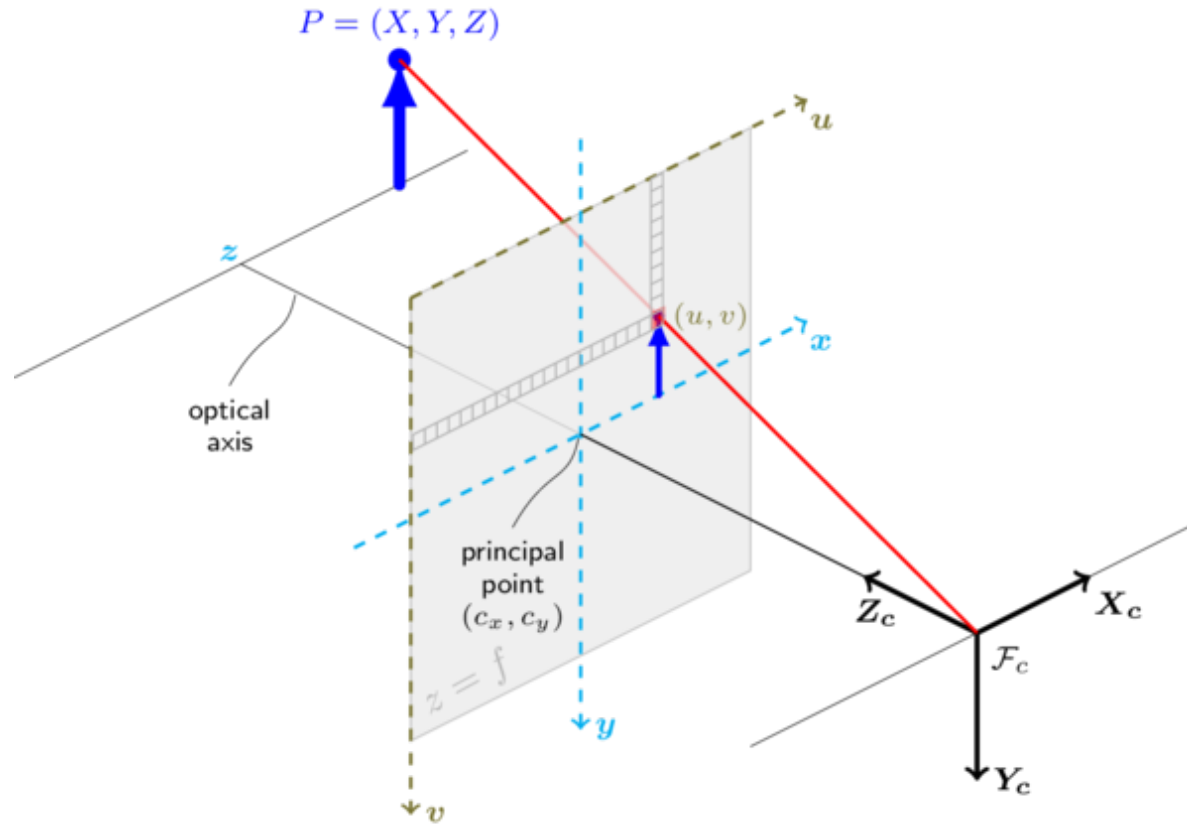
**33 34**

35 36

37



## Dírková kamera ~ středové promítání



$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \left( \mathbf{R} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \mathbf{t} \right)$$

$$\lambda \mathbf{u} = \mathbf{K}(\mathbf{R}\mathbf{x} + \mathbf{t})$$

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	

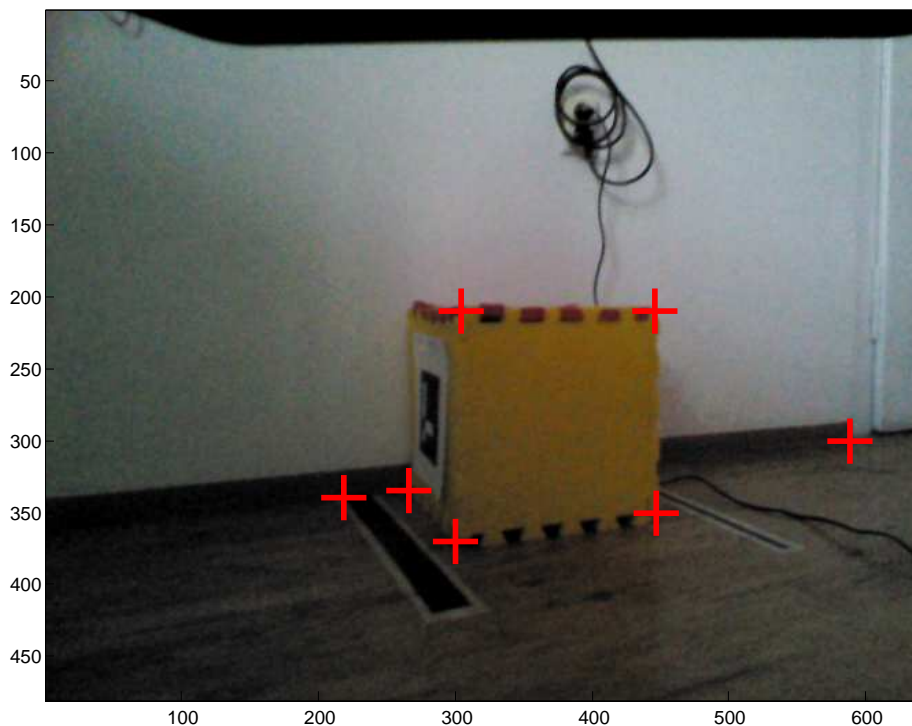


## Promítání bodů v obraze do globální souřadné soustavy

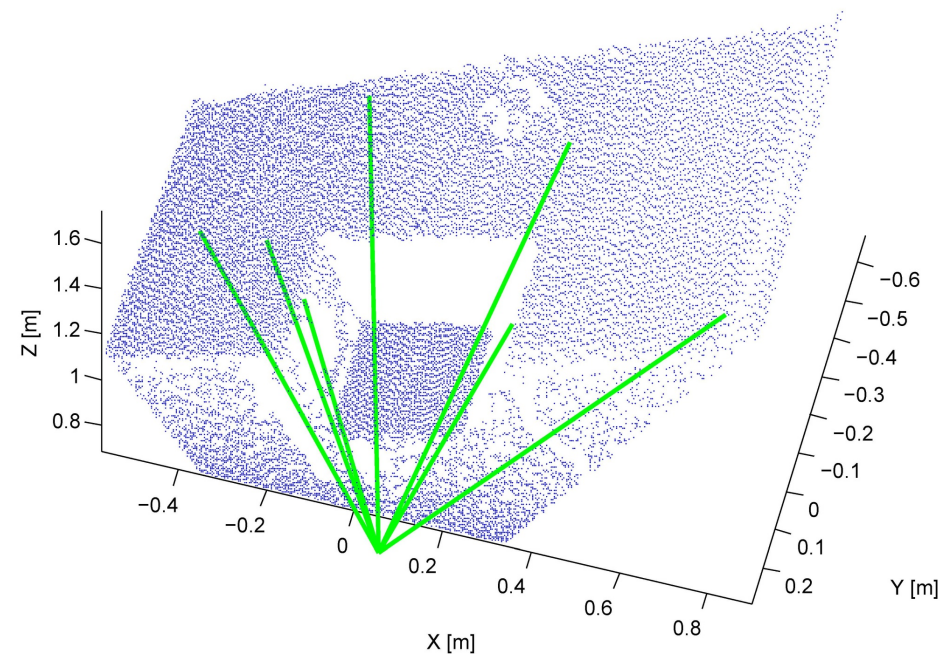
- ◆ Bodu v obraze odpovídá paprsek v prostoru
- ◆ Paprsek je přímka popsána bodem  $\mathbf{x}$  a směrovým vektorem  $\mathbf{s}$ .

V souřadné soustavě kamery platí:  $1/\lambda \mathbf{s}_C = \mathbf{K}^{-1} \mathbf{u}$  a  $\mathbf{x}_C = [0, 0, 0]$

Po transformaci do globální souřadné soustavy':  $\mathbf{s} = \mathbf{R}^{-1} \mathbf{s}_C$  a  $\mathbf{x} = \mathbf{R}^{-1} (\mathbf{x}_C - \mathbf{t})$



Body ve snímku



Odpovídající paprsky v prostoru

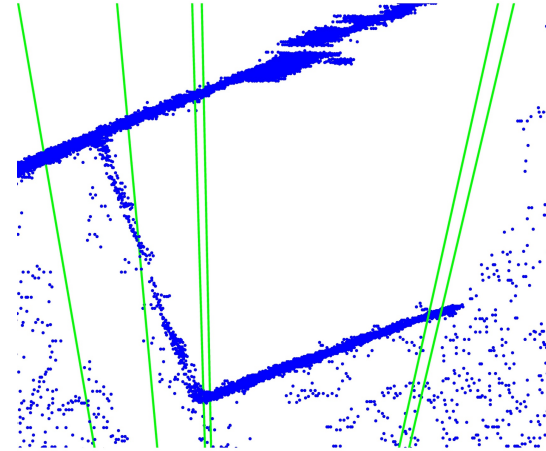
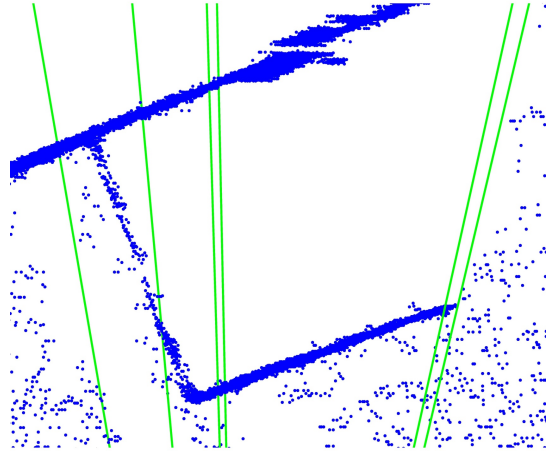
1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



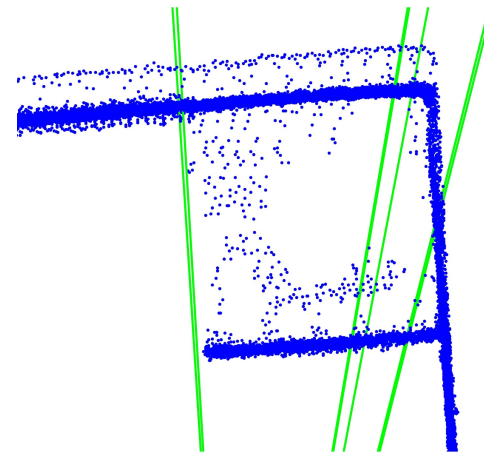
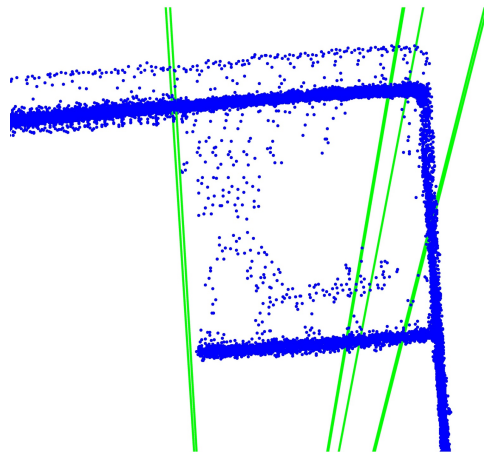
## Projekce bodů s ohledem na polohu kamer

Výrobce udávaná vzdálenost RGB a hloubkové kamery je 2,5 cm.

Rovina  
X-Z



Rovina  
Y-Z



$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, t = [0, 0, 0]$$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, t = [0.025, 0, 0]$$

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	



## Zpracování obrazu

[Sonka1993] Mialn Sonka, Vaclav Hlavac a Roger Boyle. *Image Processing, Analysis and Machine Vision*. ISBN: 978-0-412-45570-4, Springer US, 1993.

[Zanuttigh2016] Pietro Zanuttigh, Giulio Marin, Carlo Dal Mutto, Fabio Dominio, Ludovico Minto a Guido Maria Cortelazzo. *Time-of-Flight and Structured Light Depth Cameras: Technology and Applications*. ISBN: 978-3-319-30971-2, Springer, 2016.

## Houghova transformace

[DudaHart1972] R.O. Duda a P.E. Hart. *Use of the Hough transformation to detect lines and curves in pictures*. Communications of the ACM, 15(1):11–15, 1972.

[Ballard1981] D.H. Ballard. *Generalizing the Hough transform to detect arbitrary shapes*. Pattern Recognition, ISSN: 0031-3203, 13(2):111 - 122, 1981

1 2

3 4

5 6

7 8

9 10

11 12

13 14

15 16

17 18

19 20

21 22

23 24

25 26

27 28

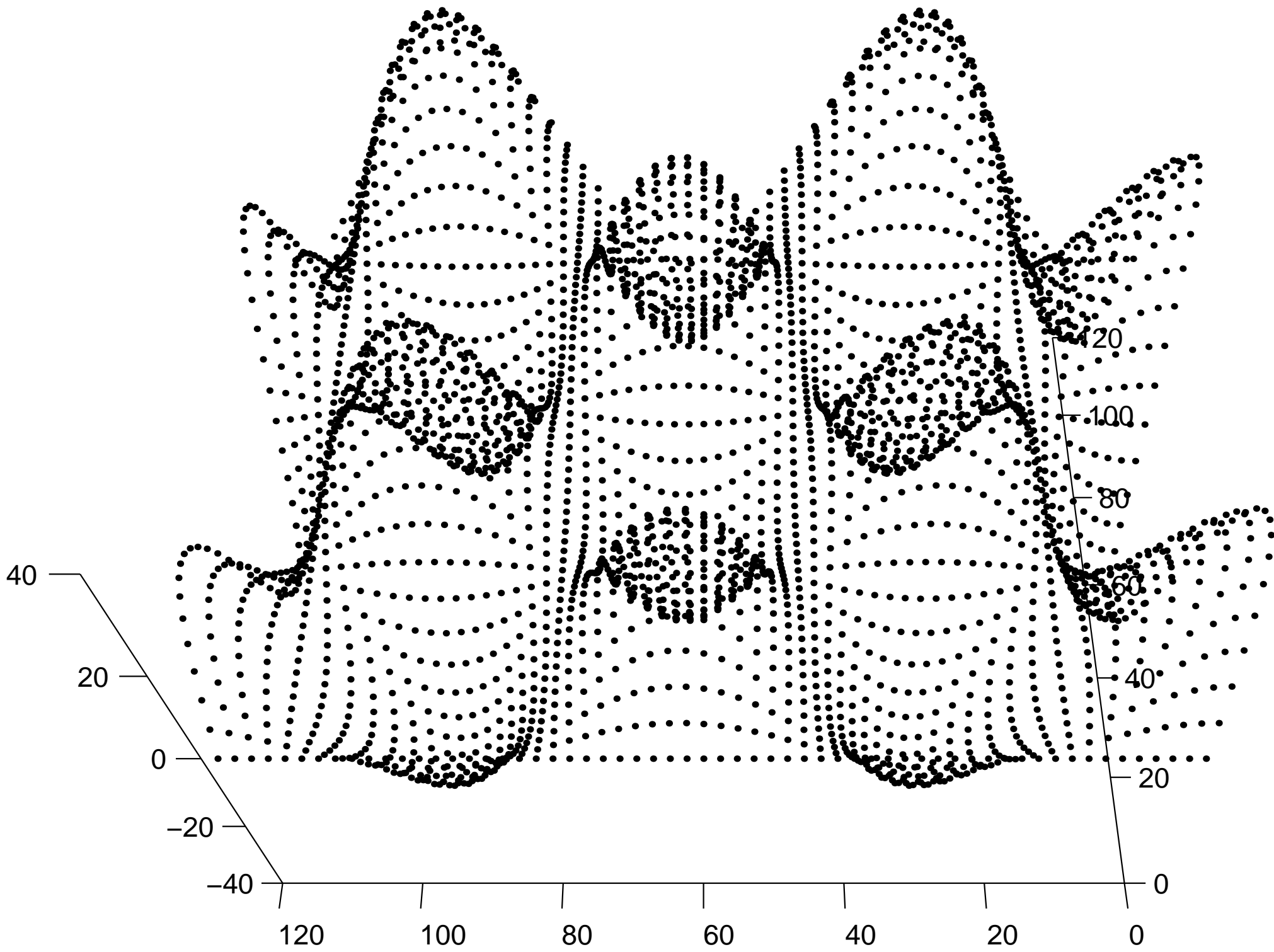
29 30

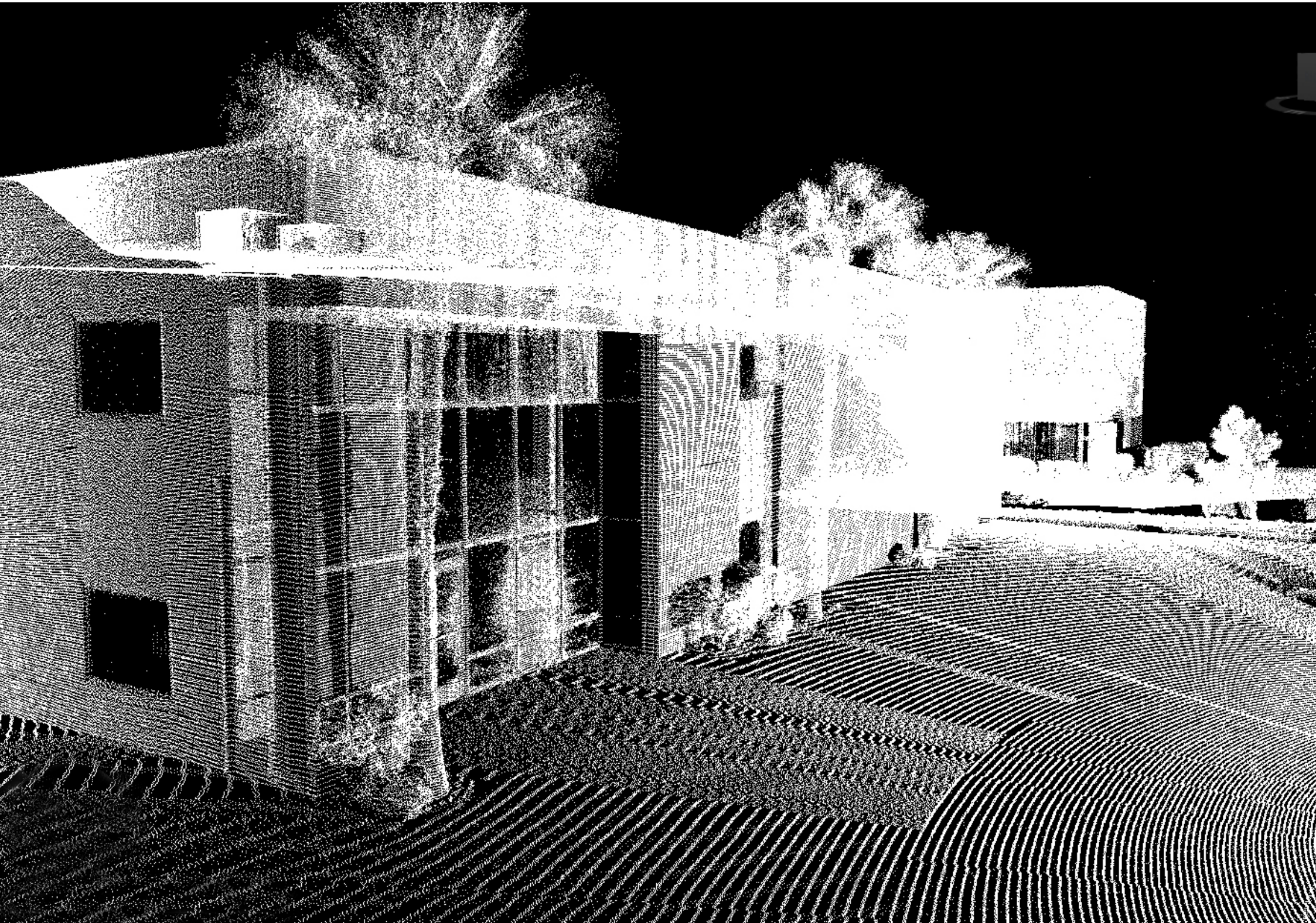
31 32

33 34

35 36

37

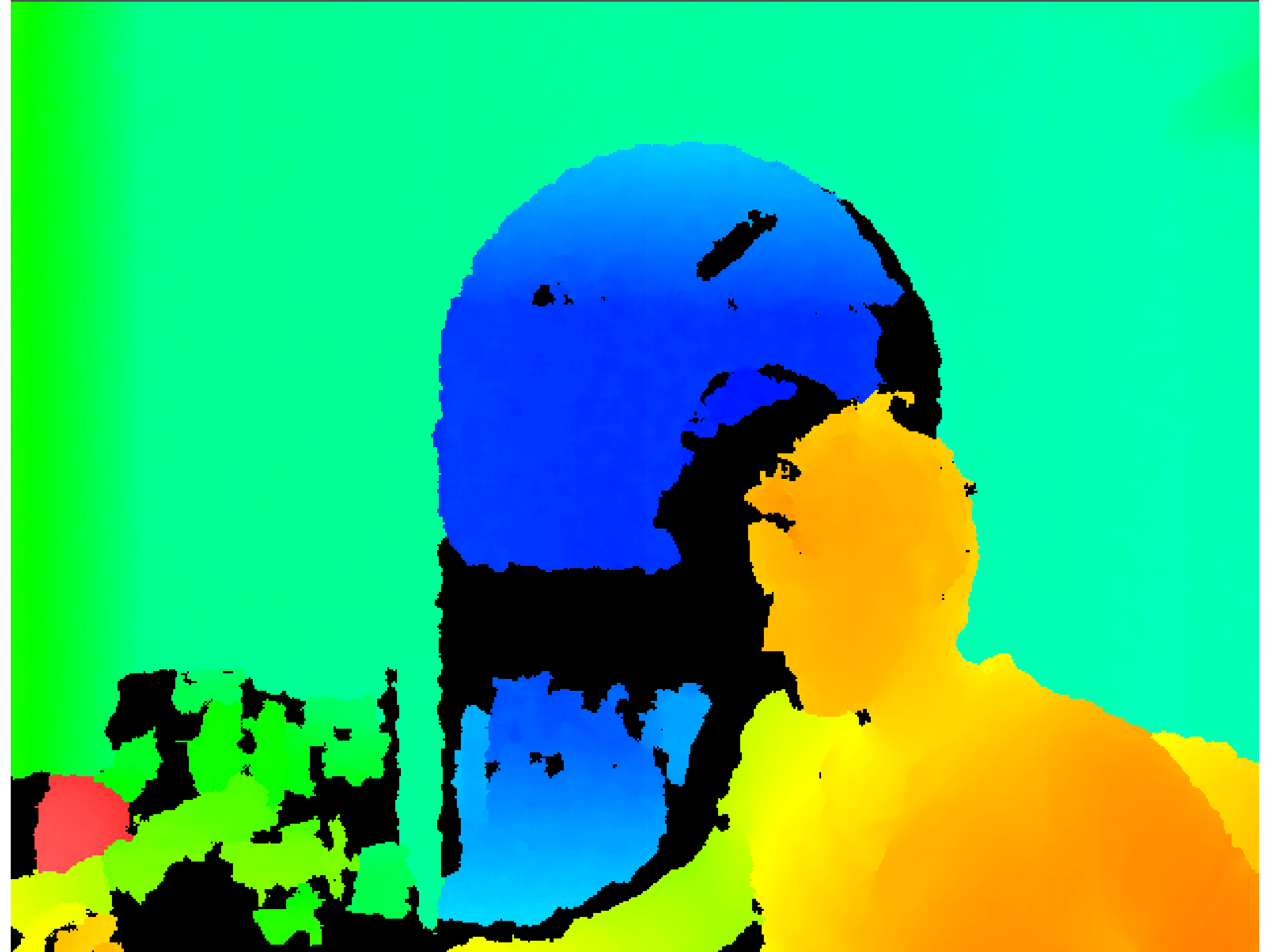


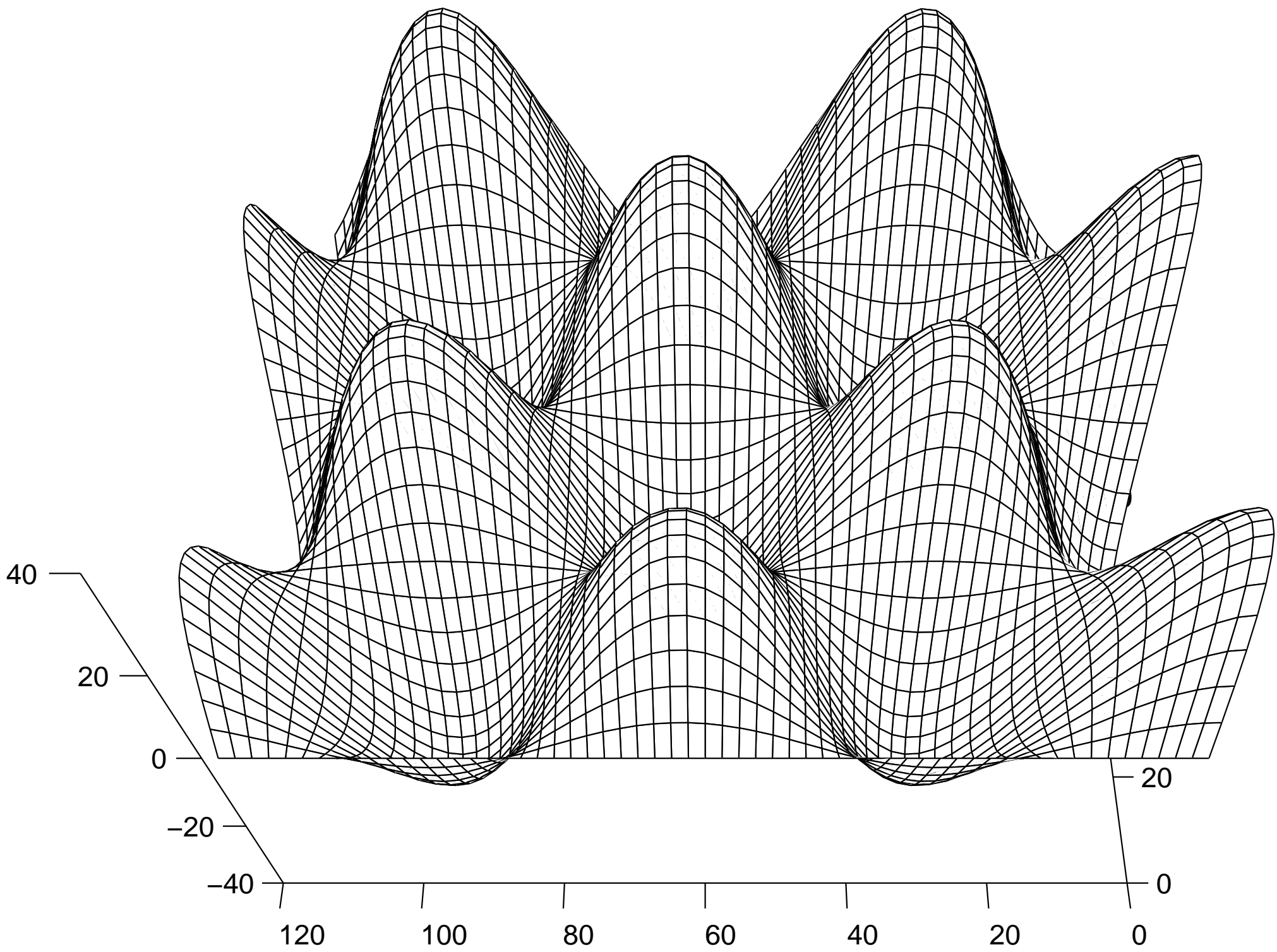


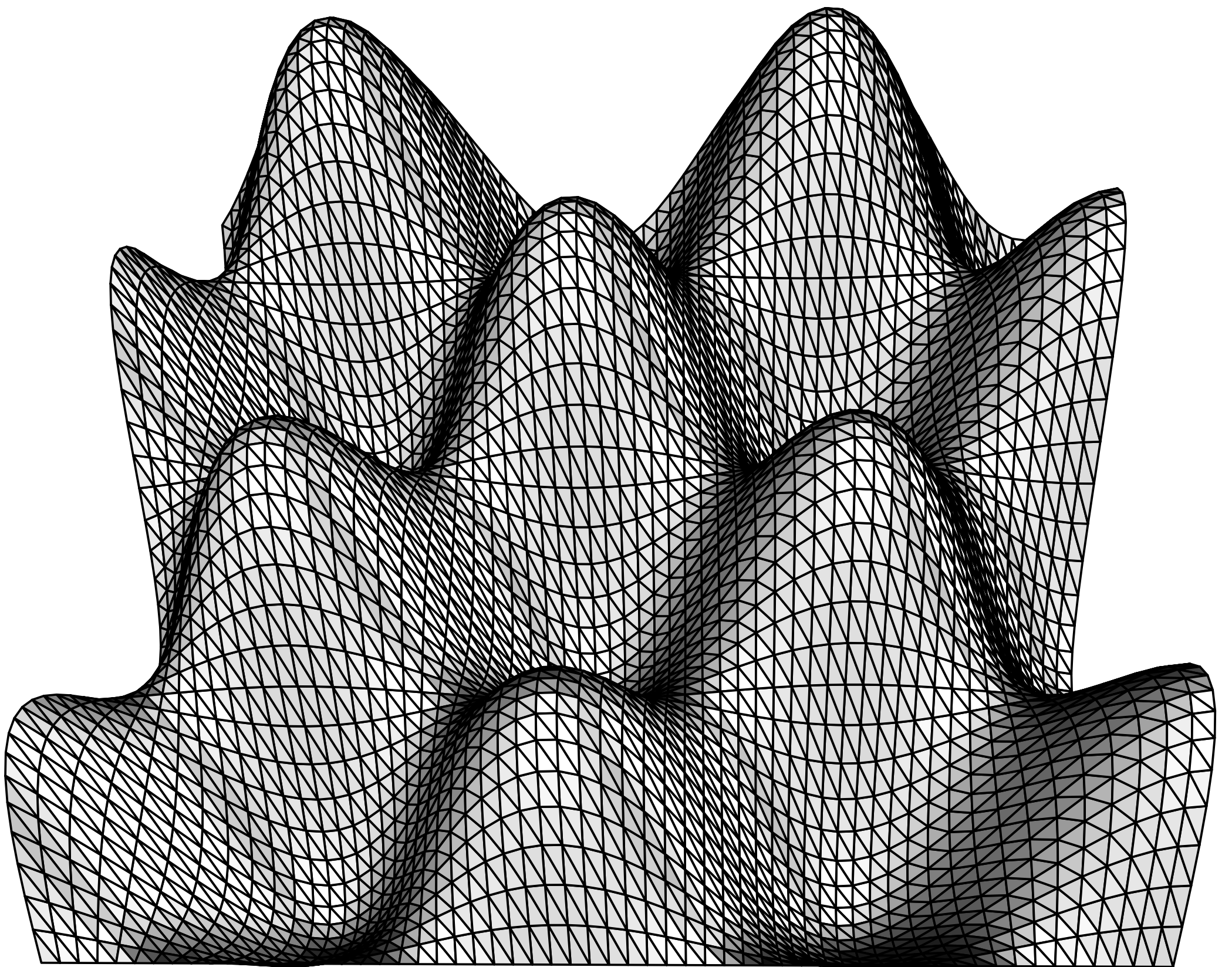




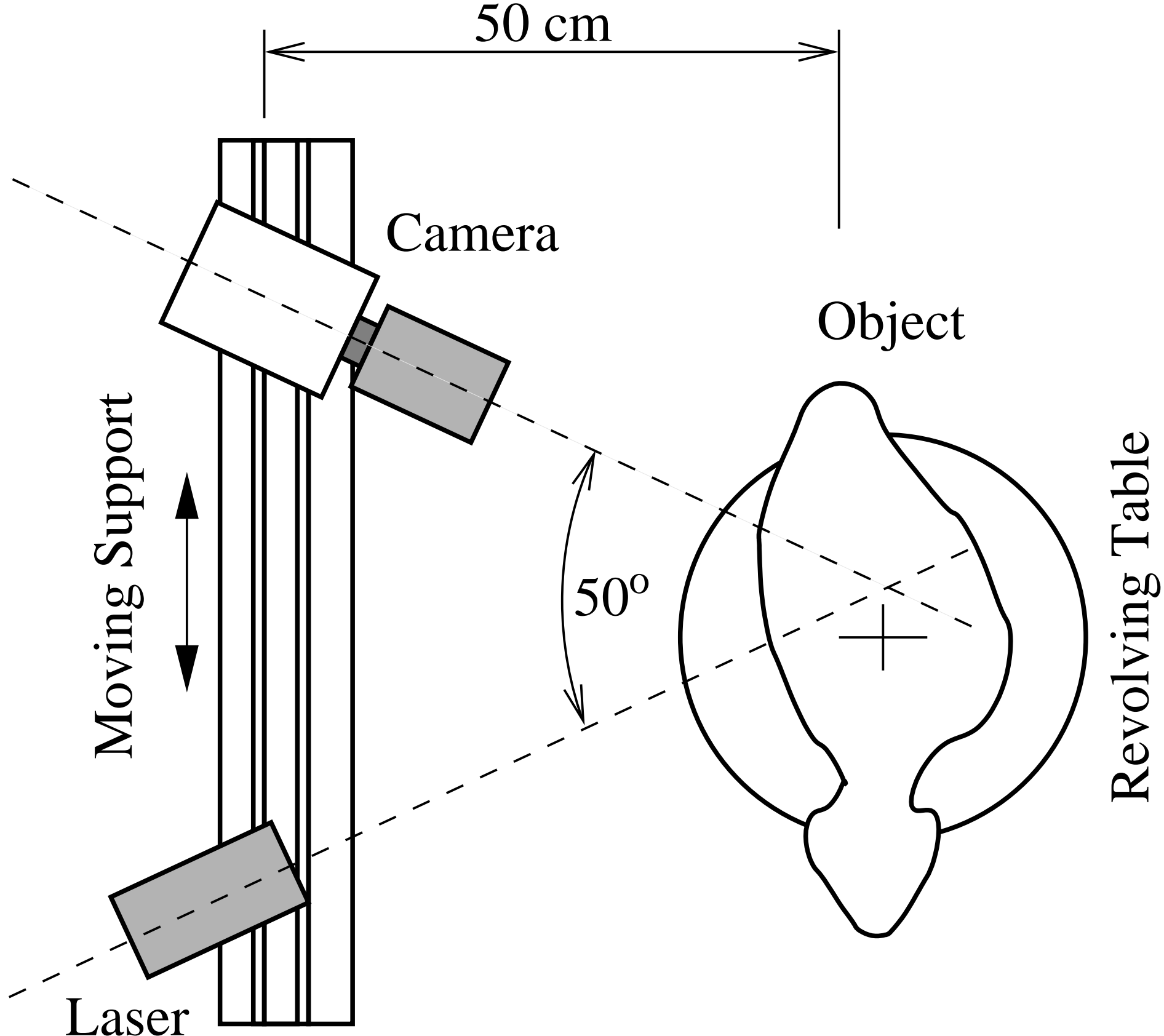


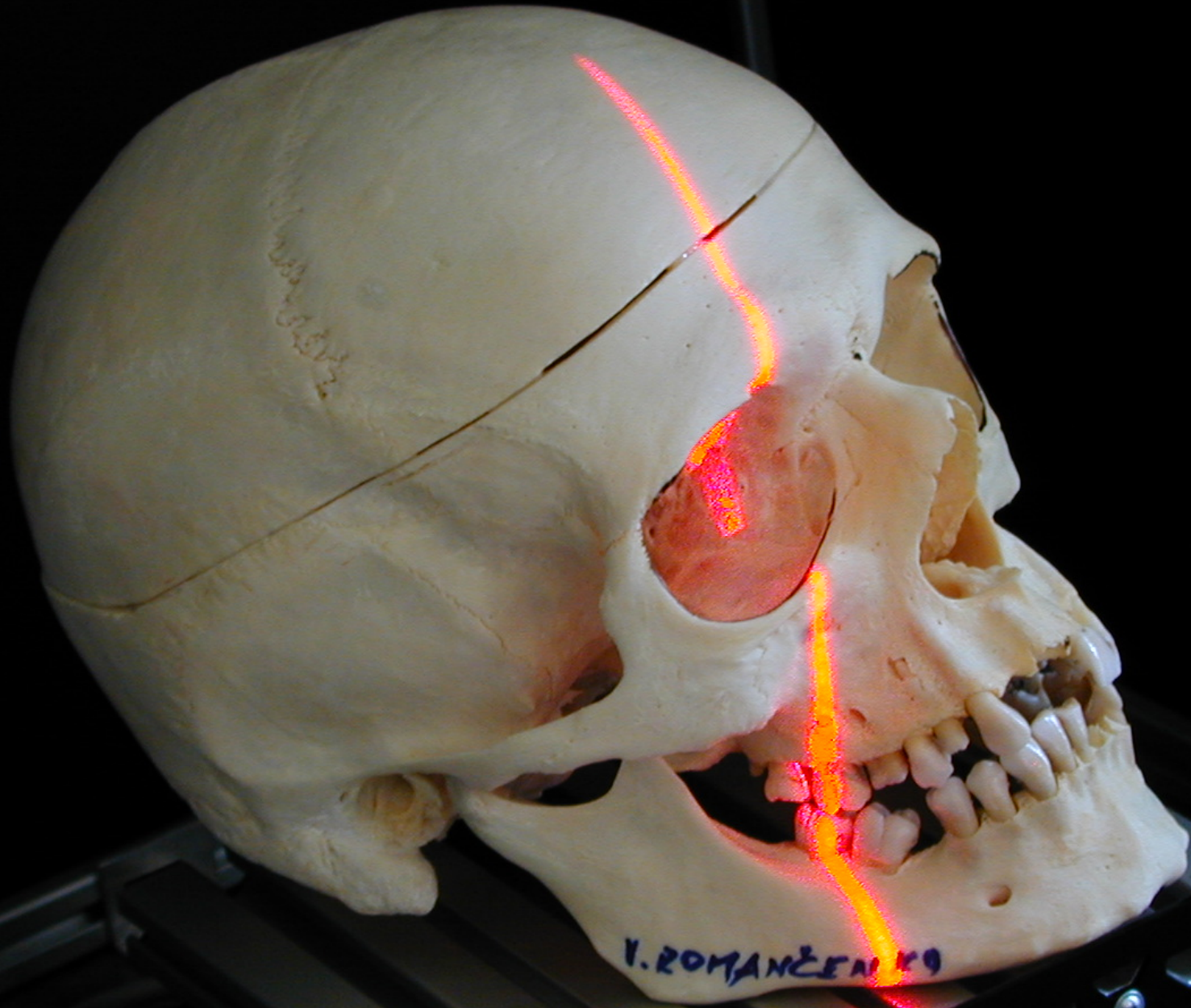








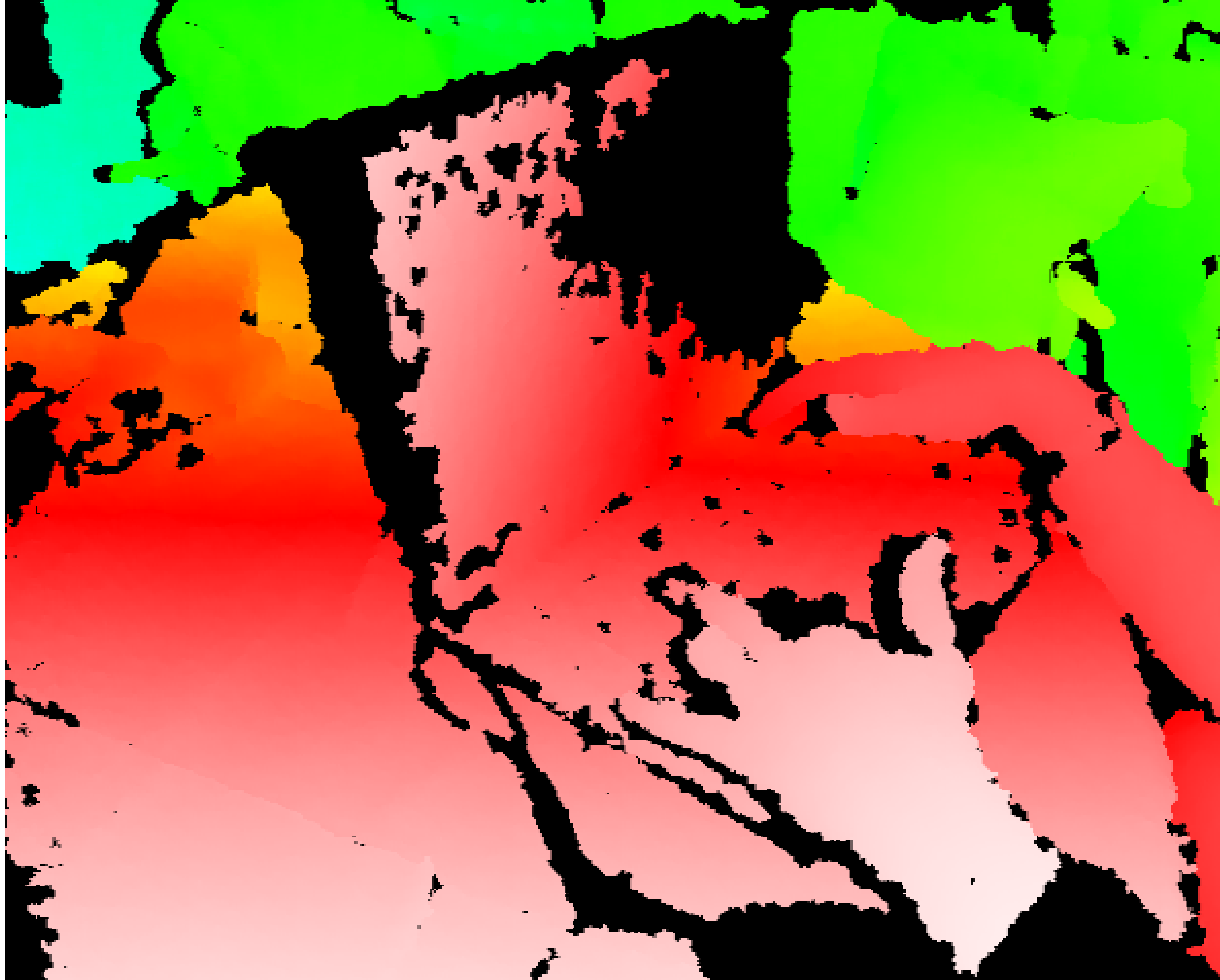


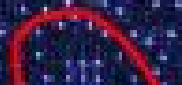
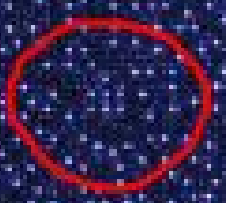
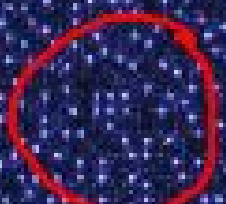
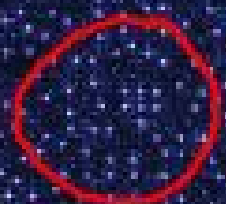


V. ROMANČENKO





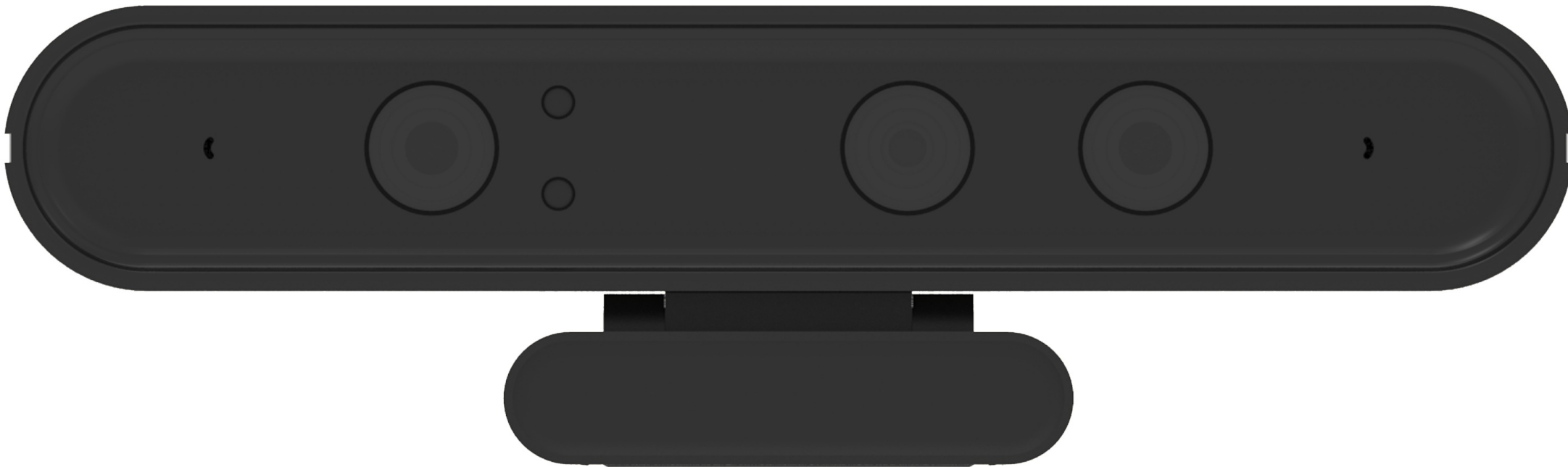


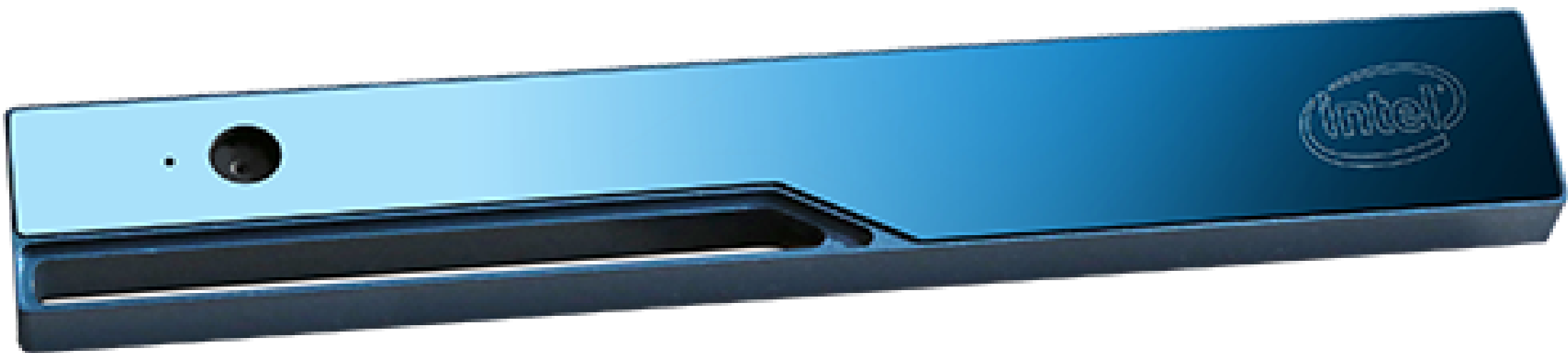


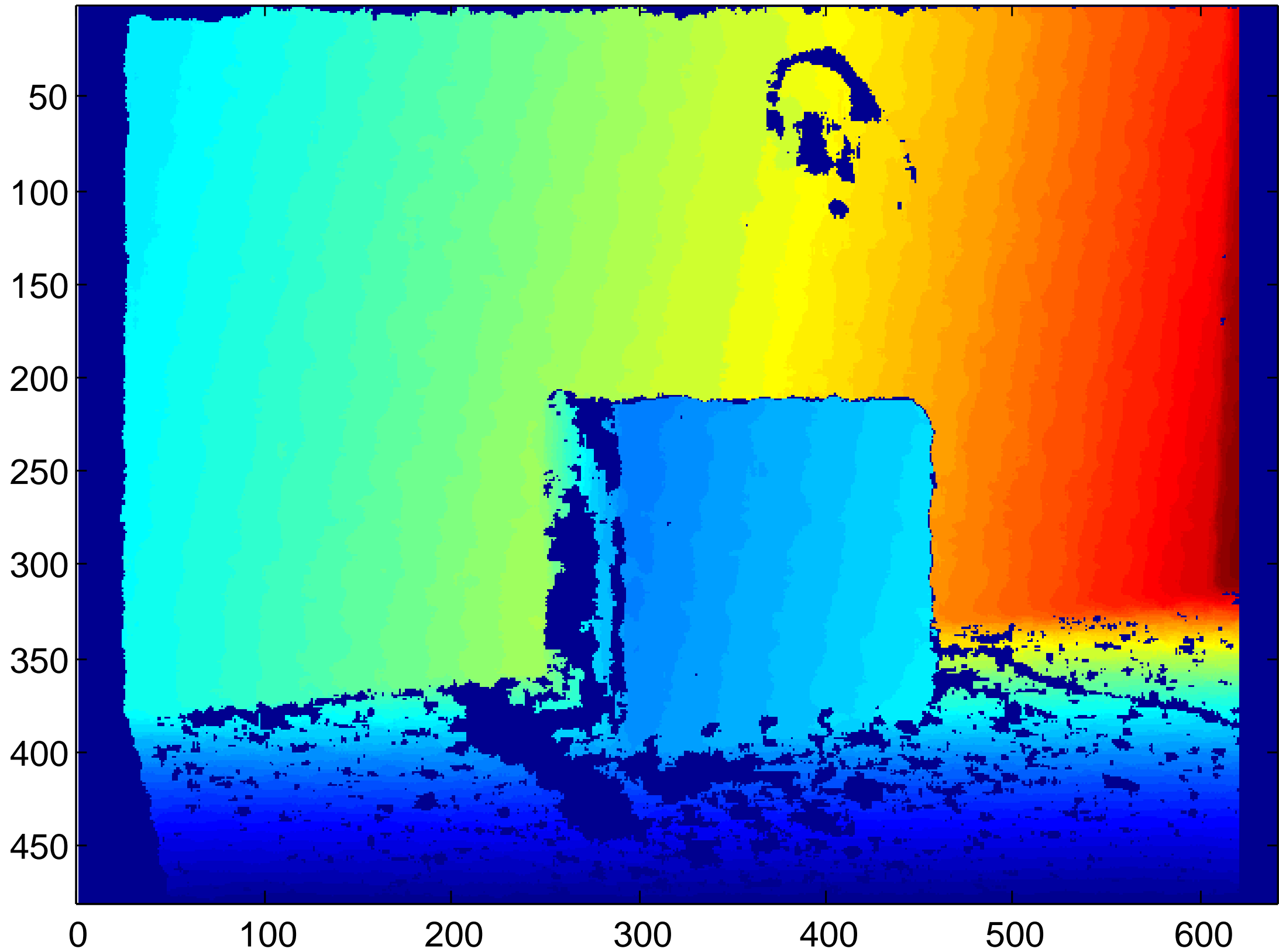


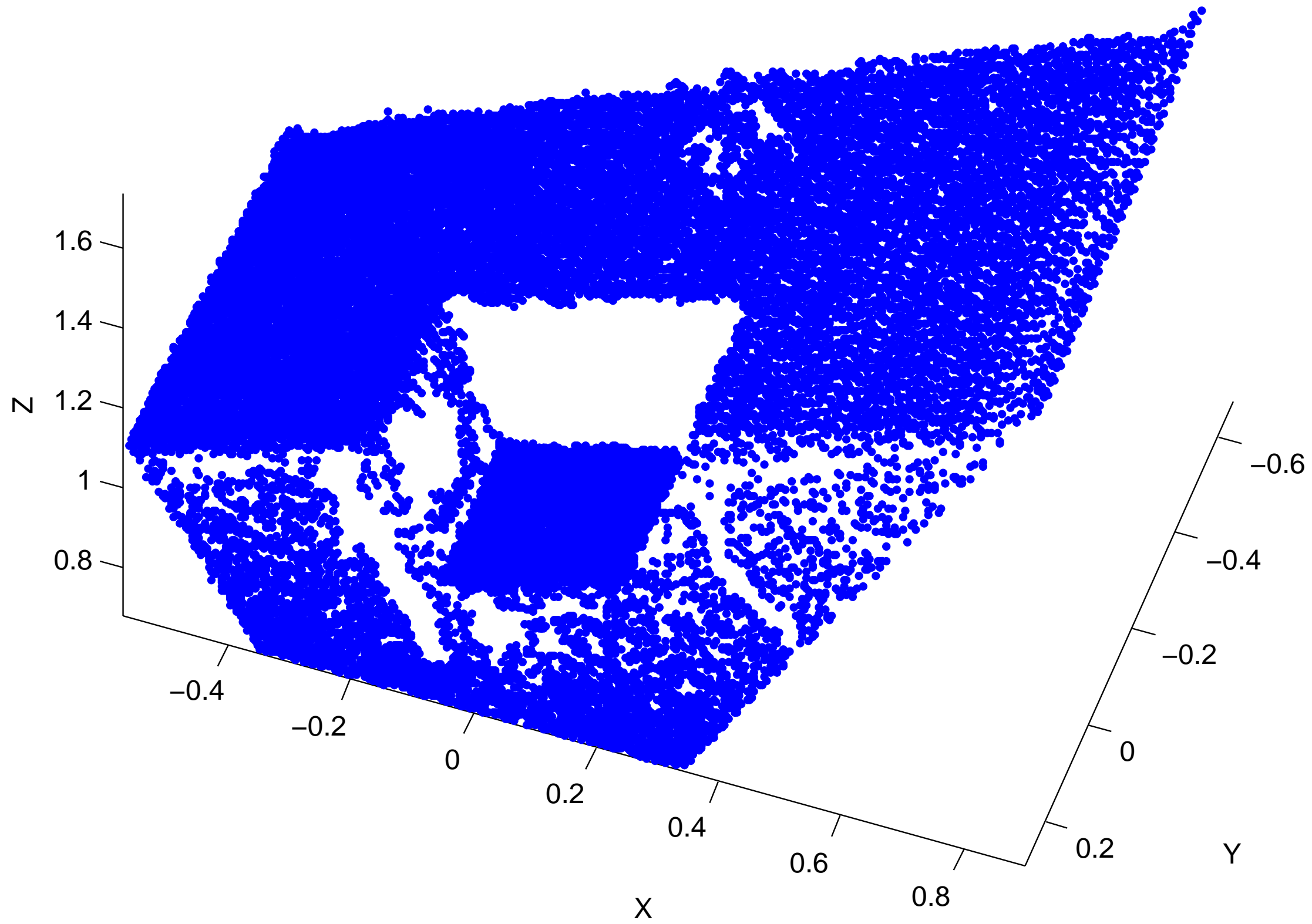
XBOX 360

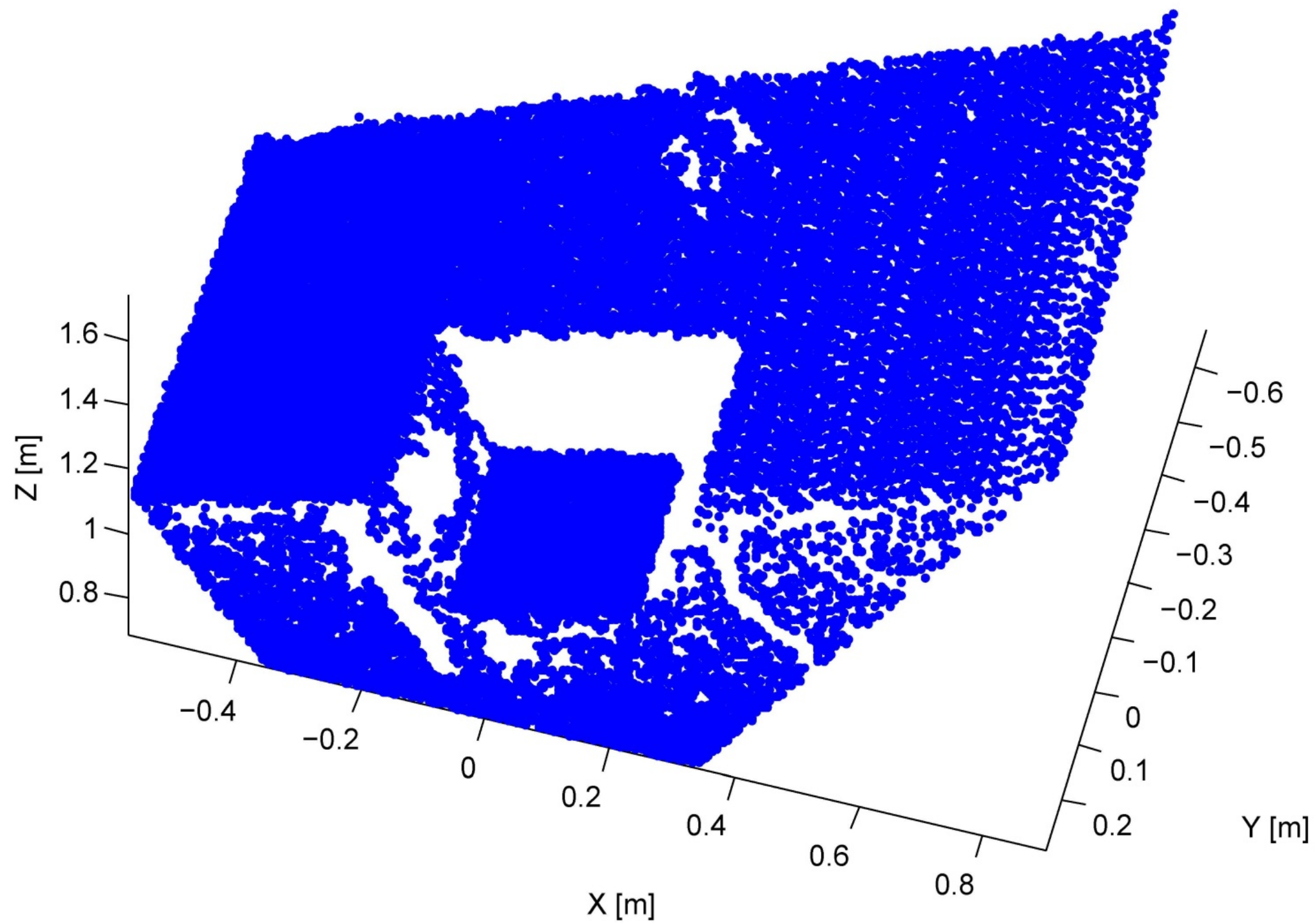
KINECT



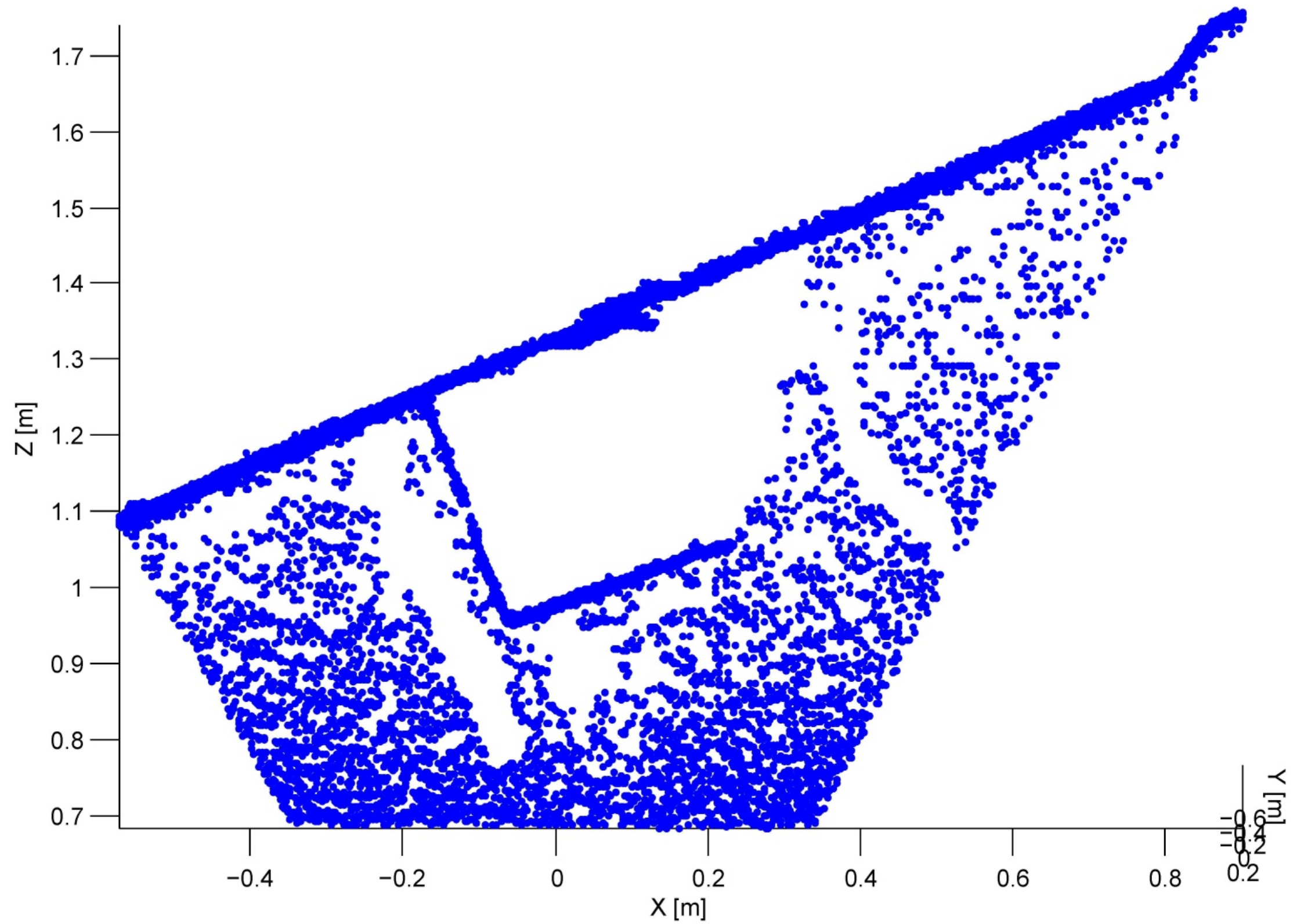


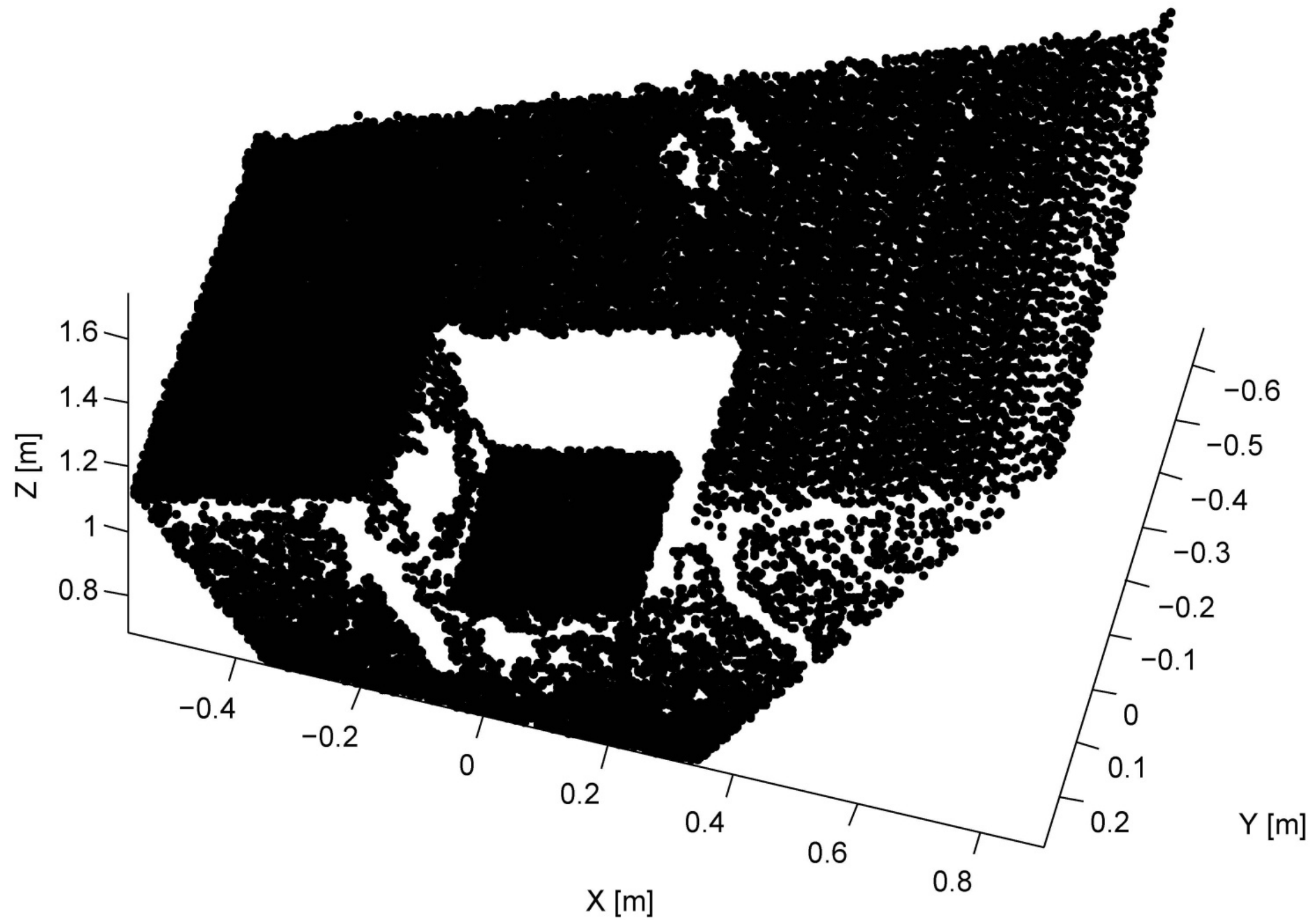


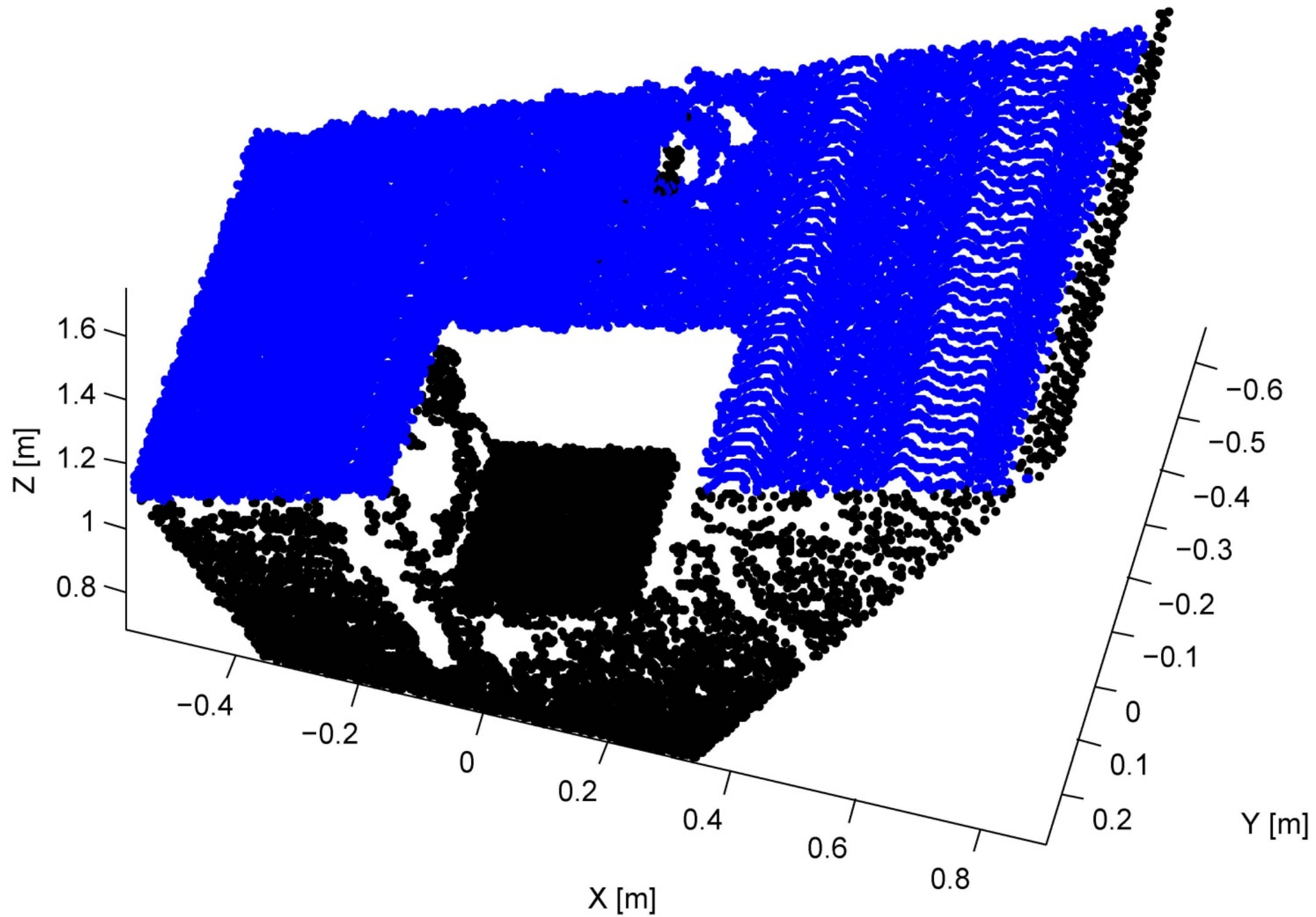


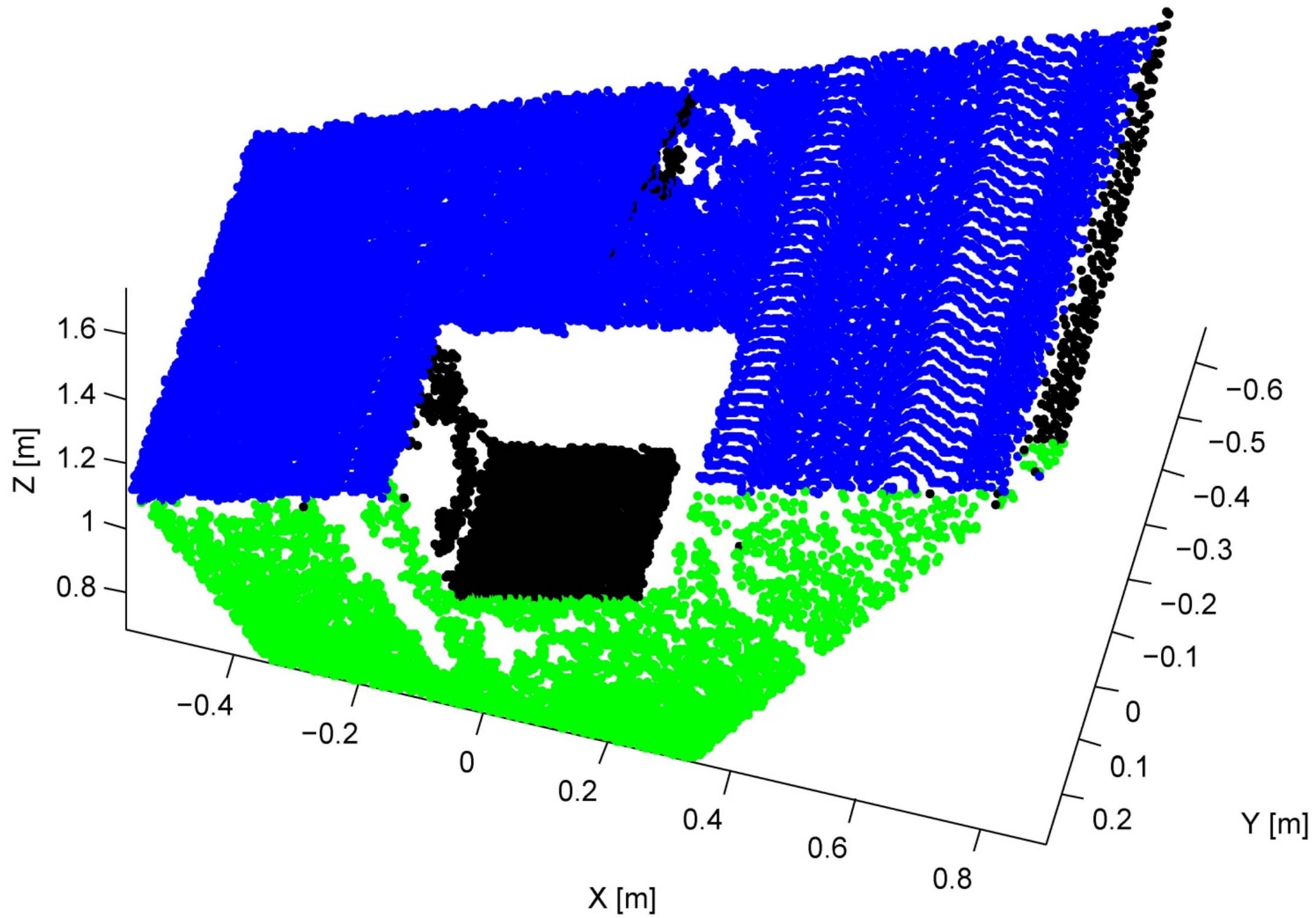


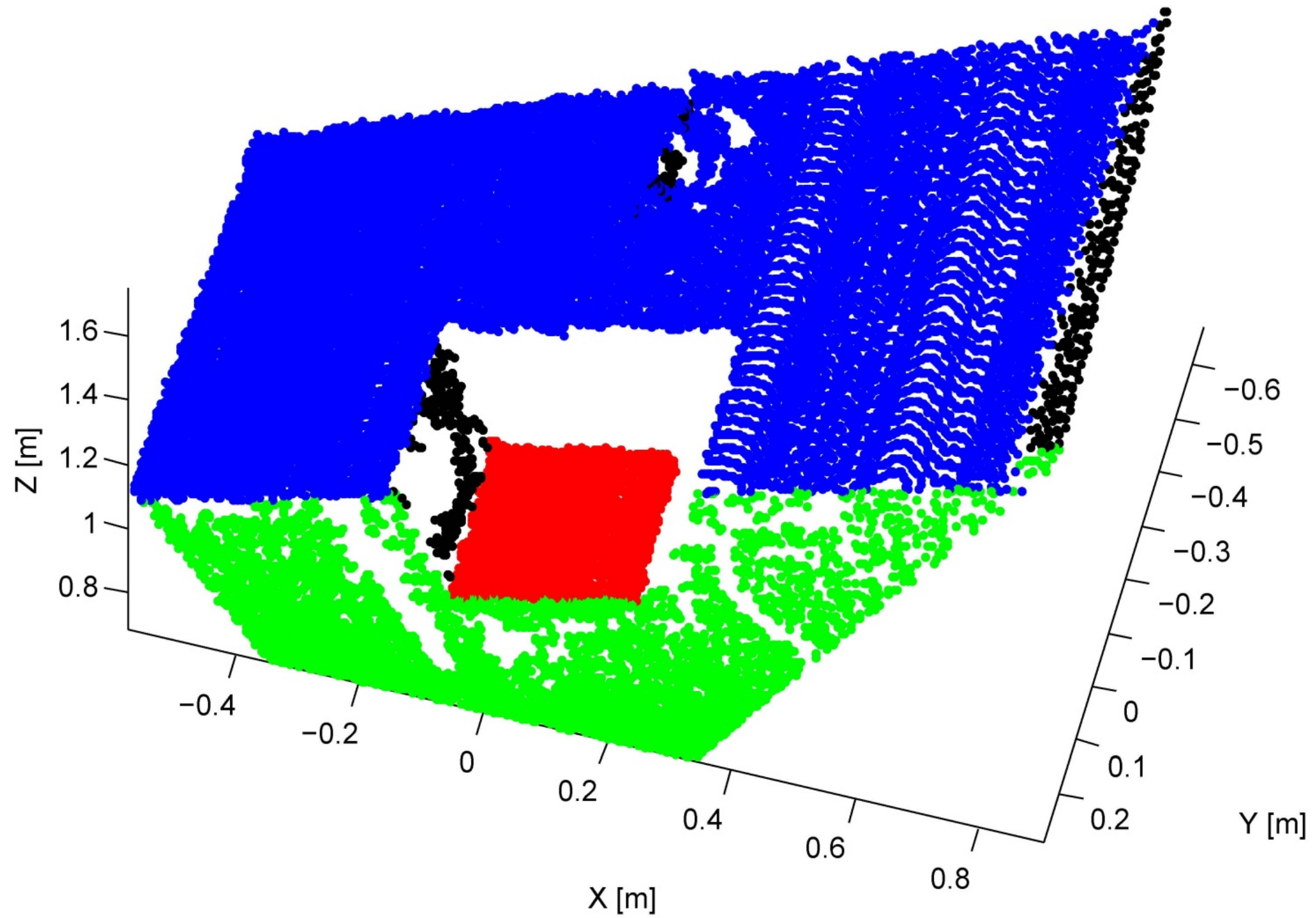


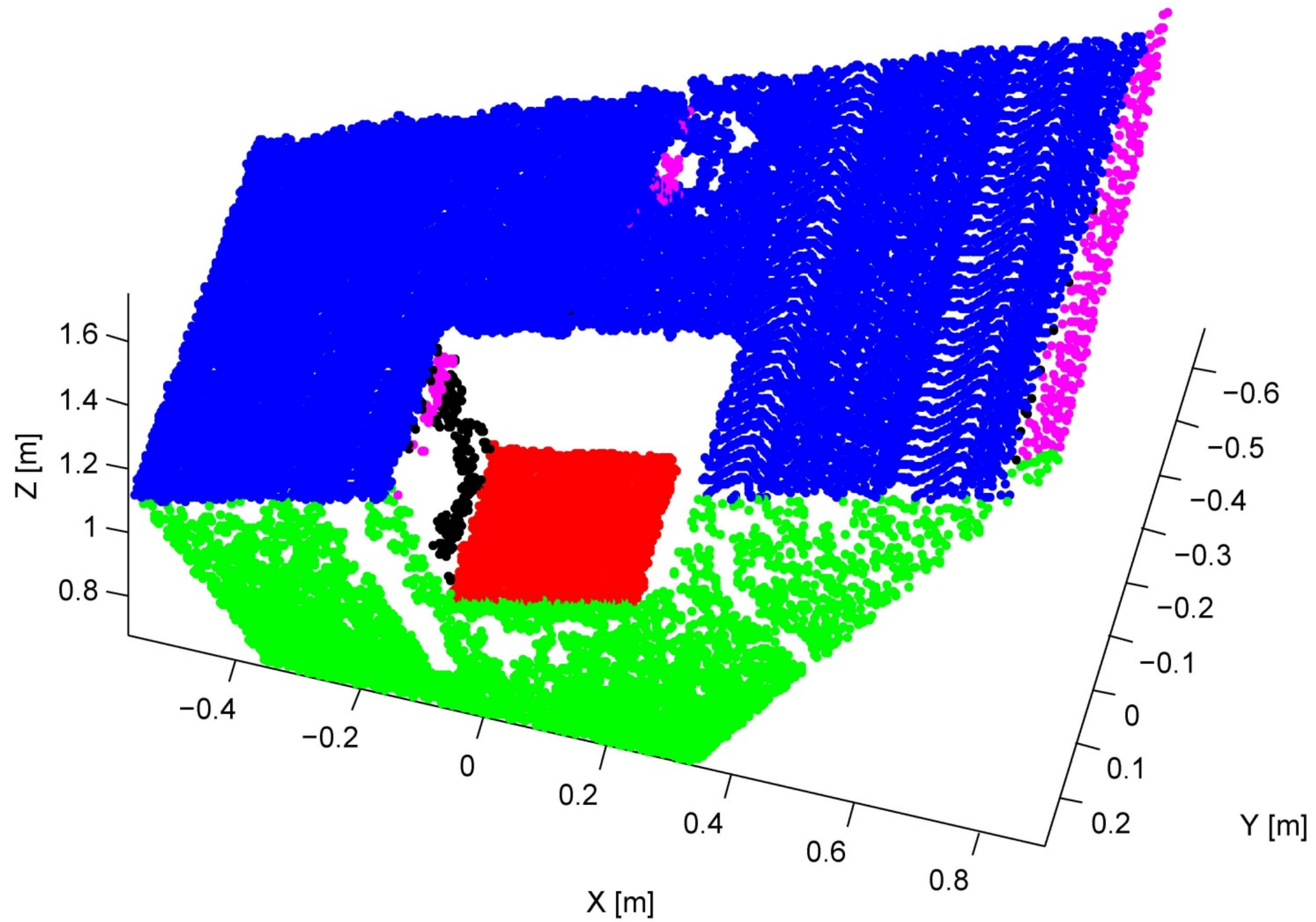


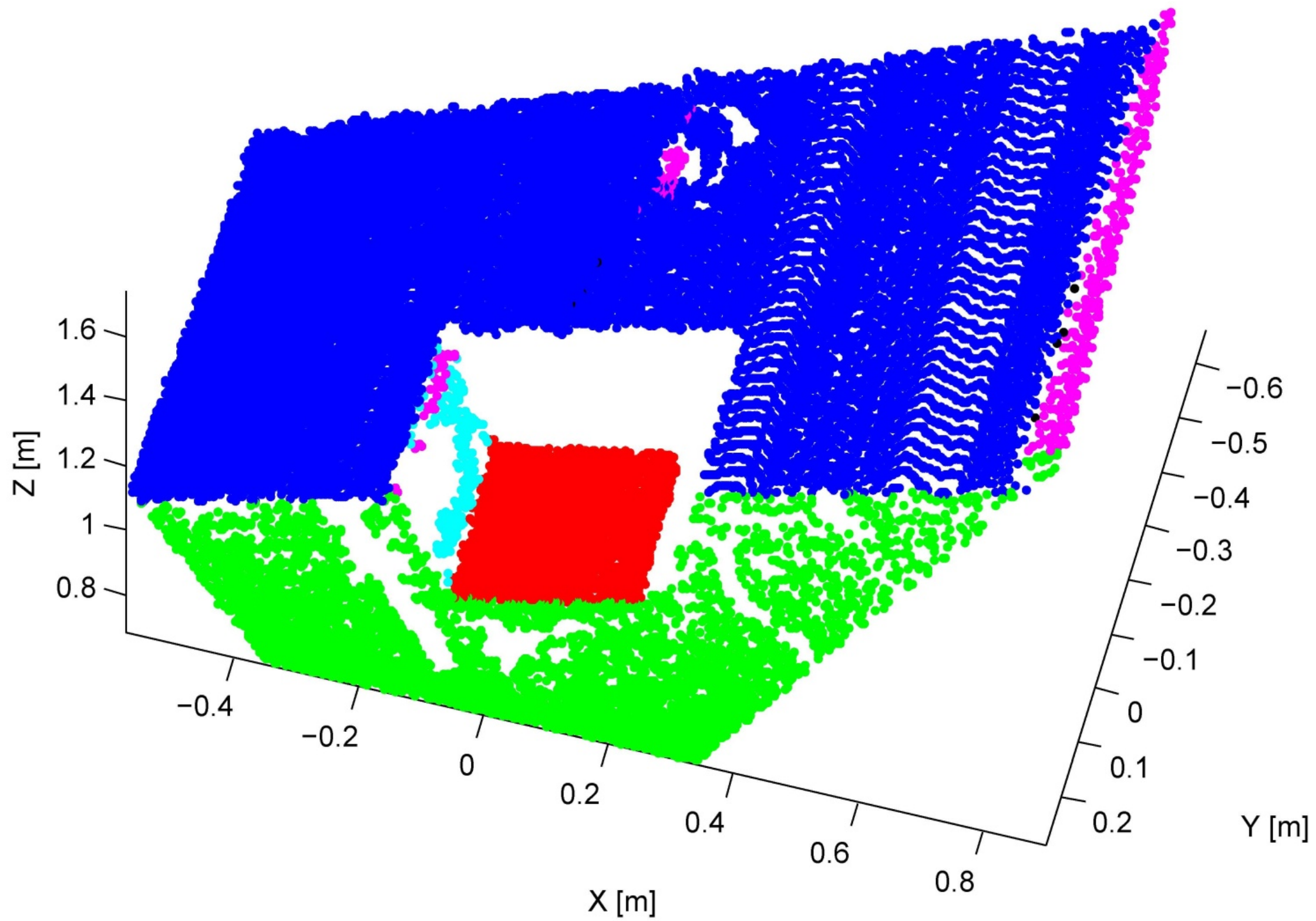


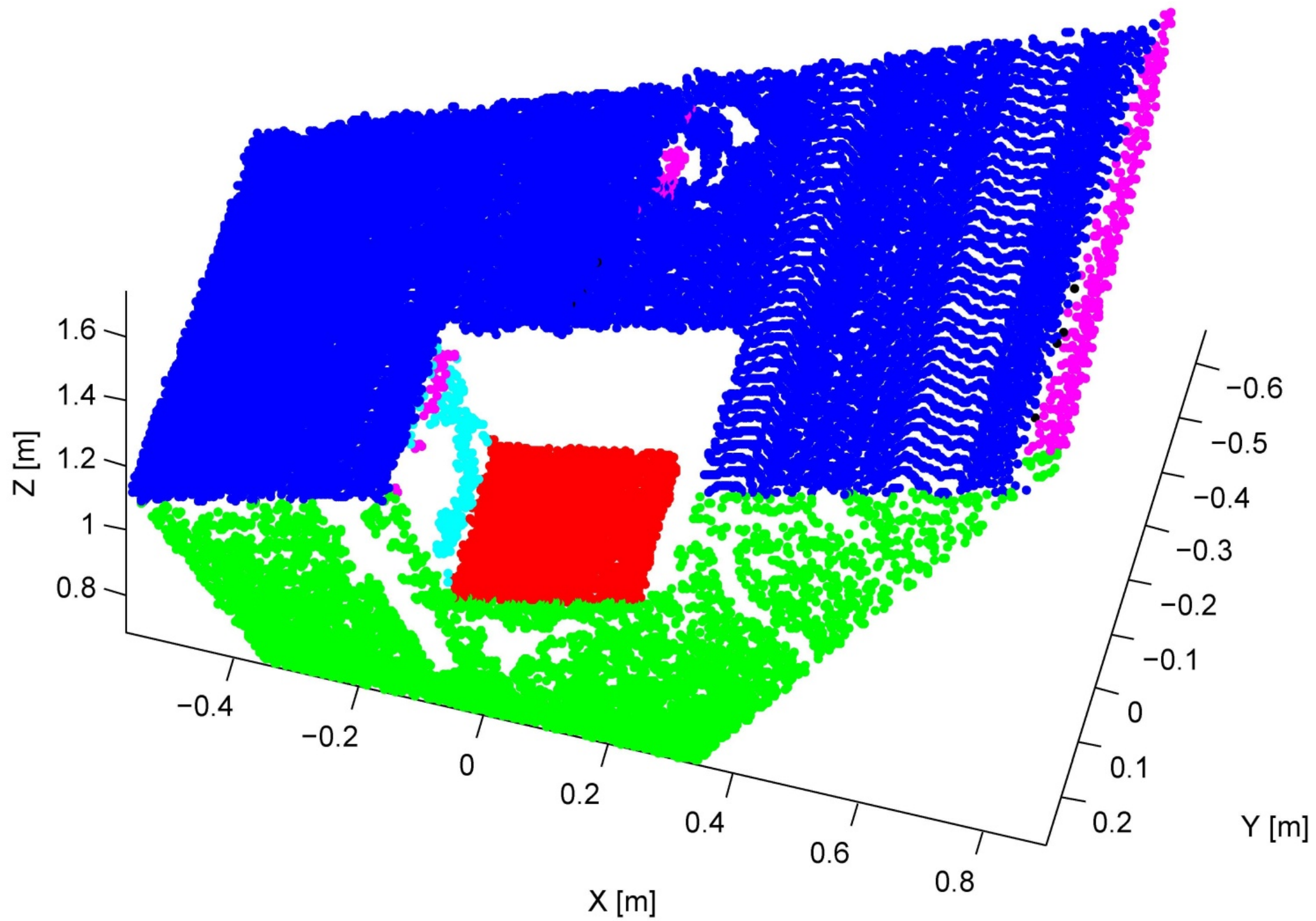




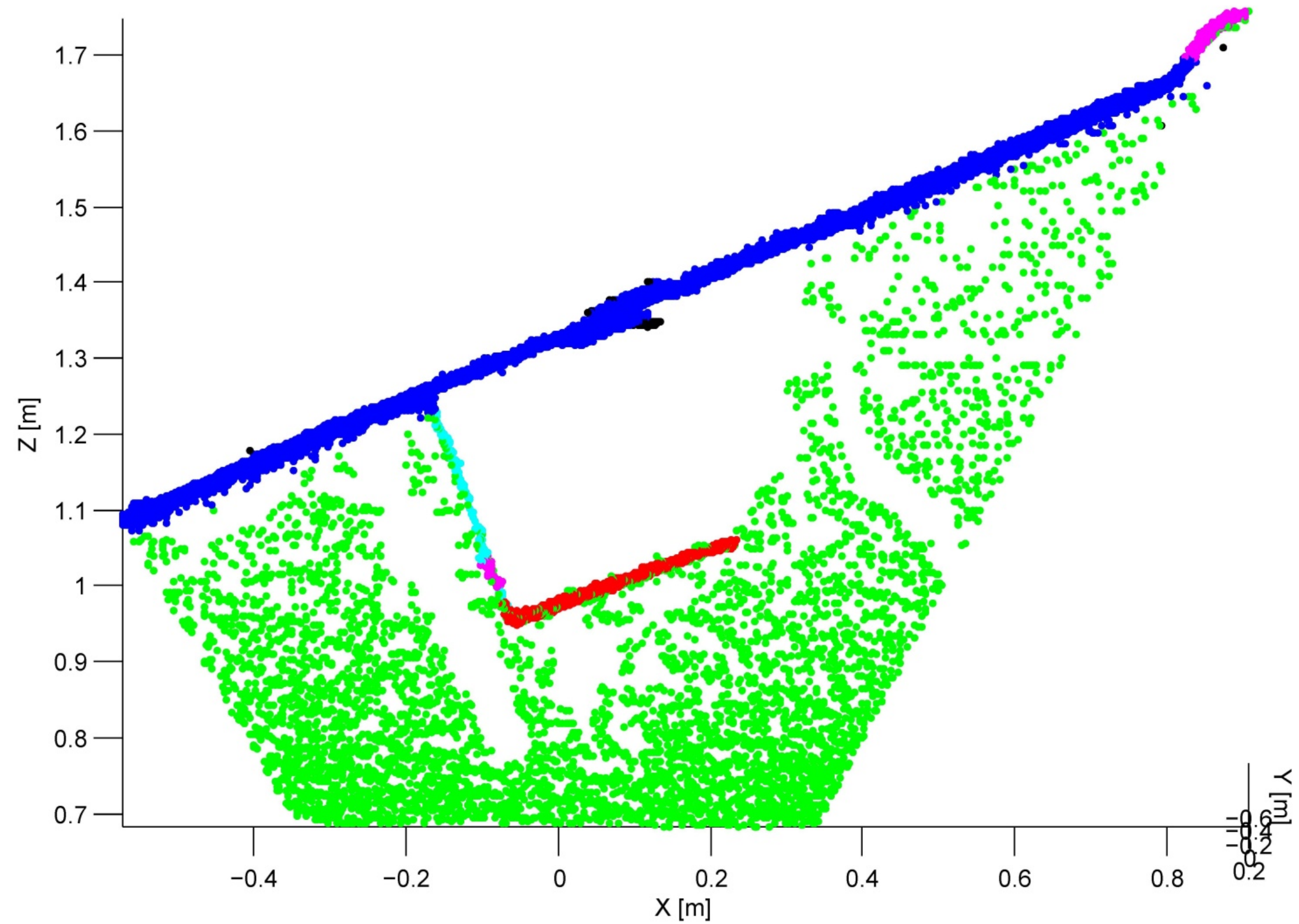












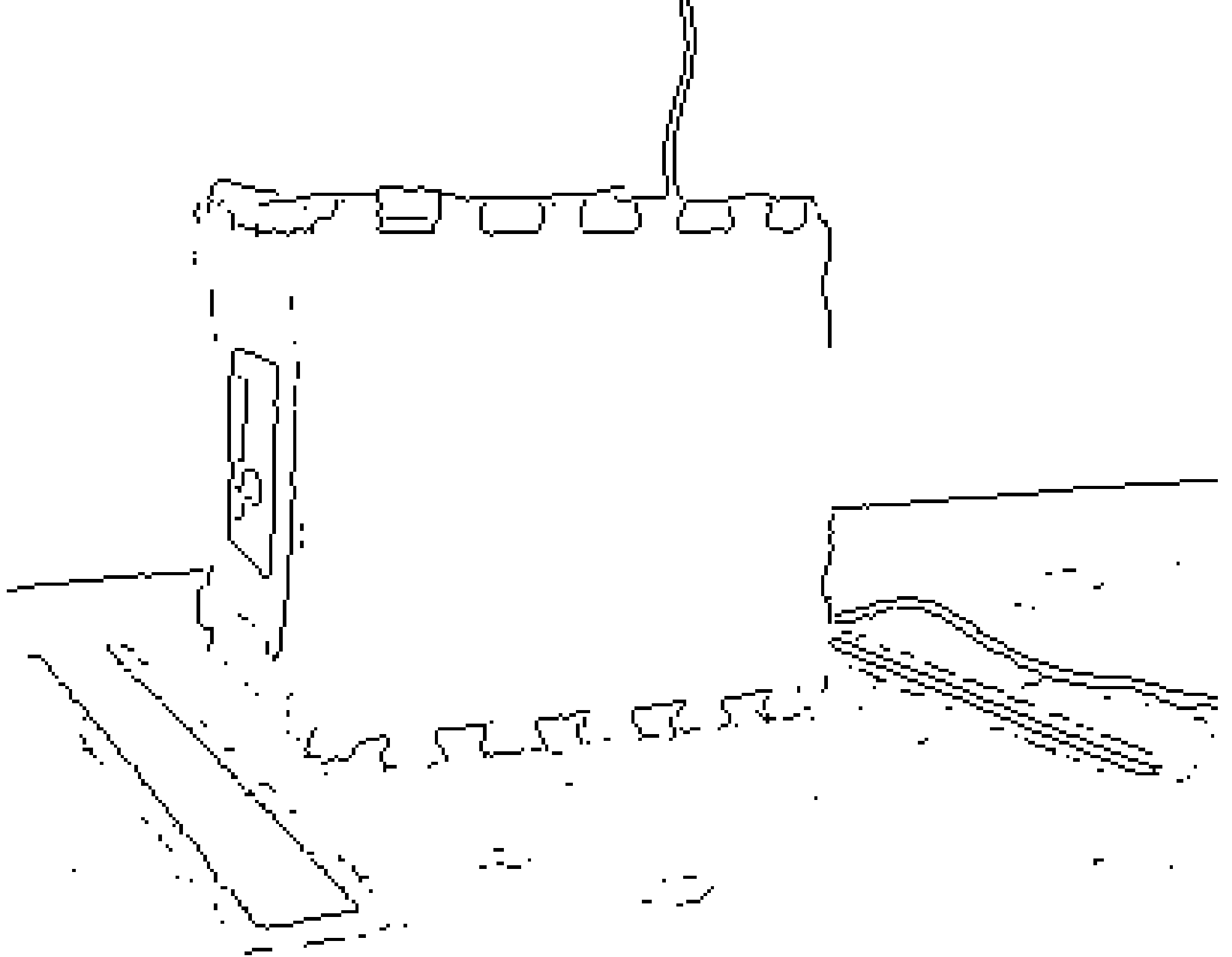


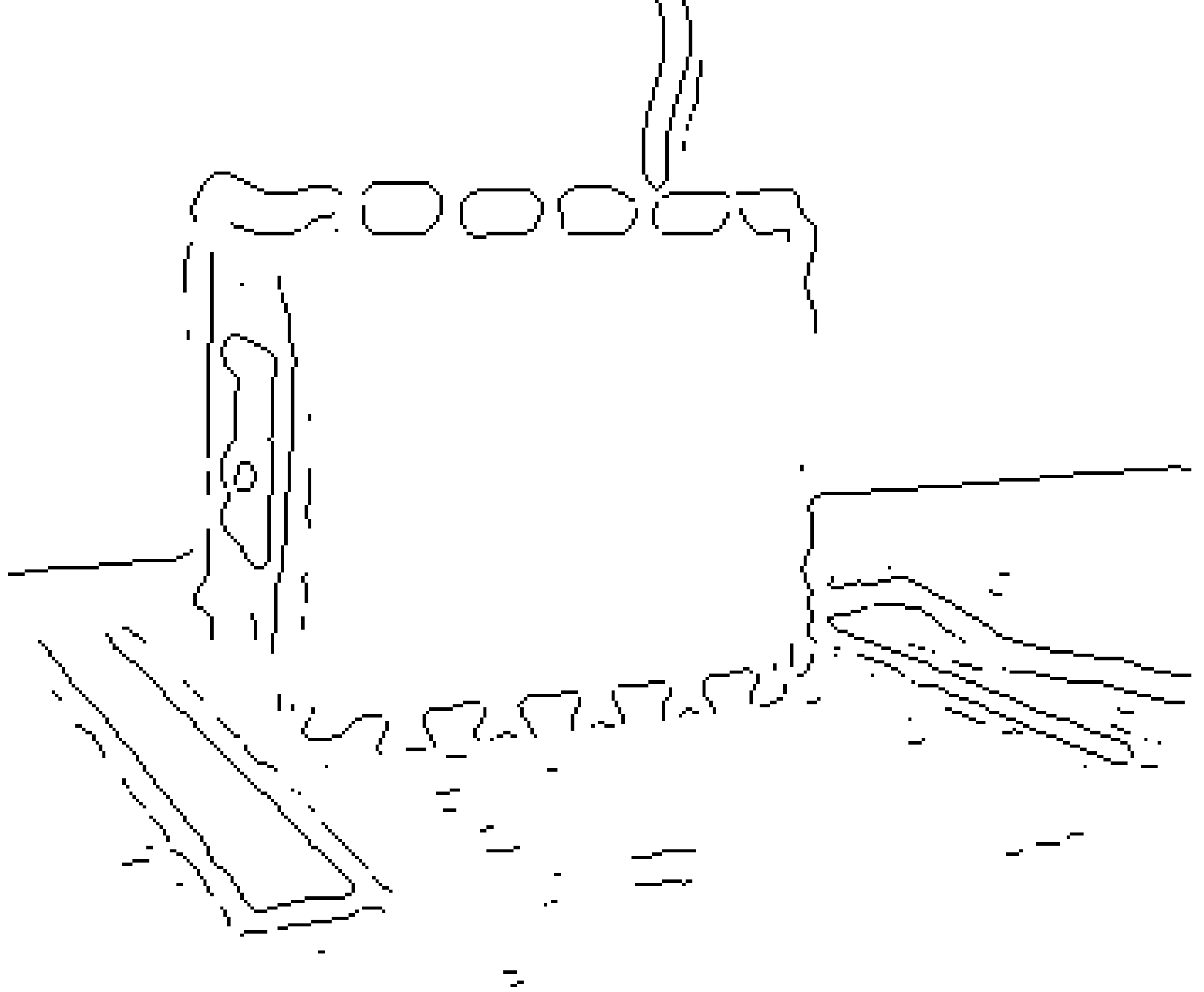




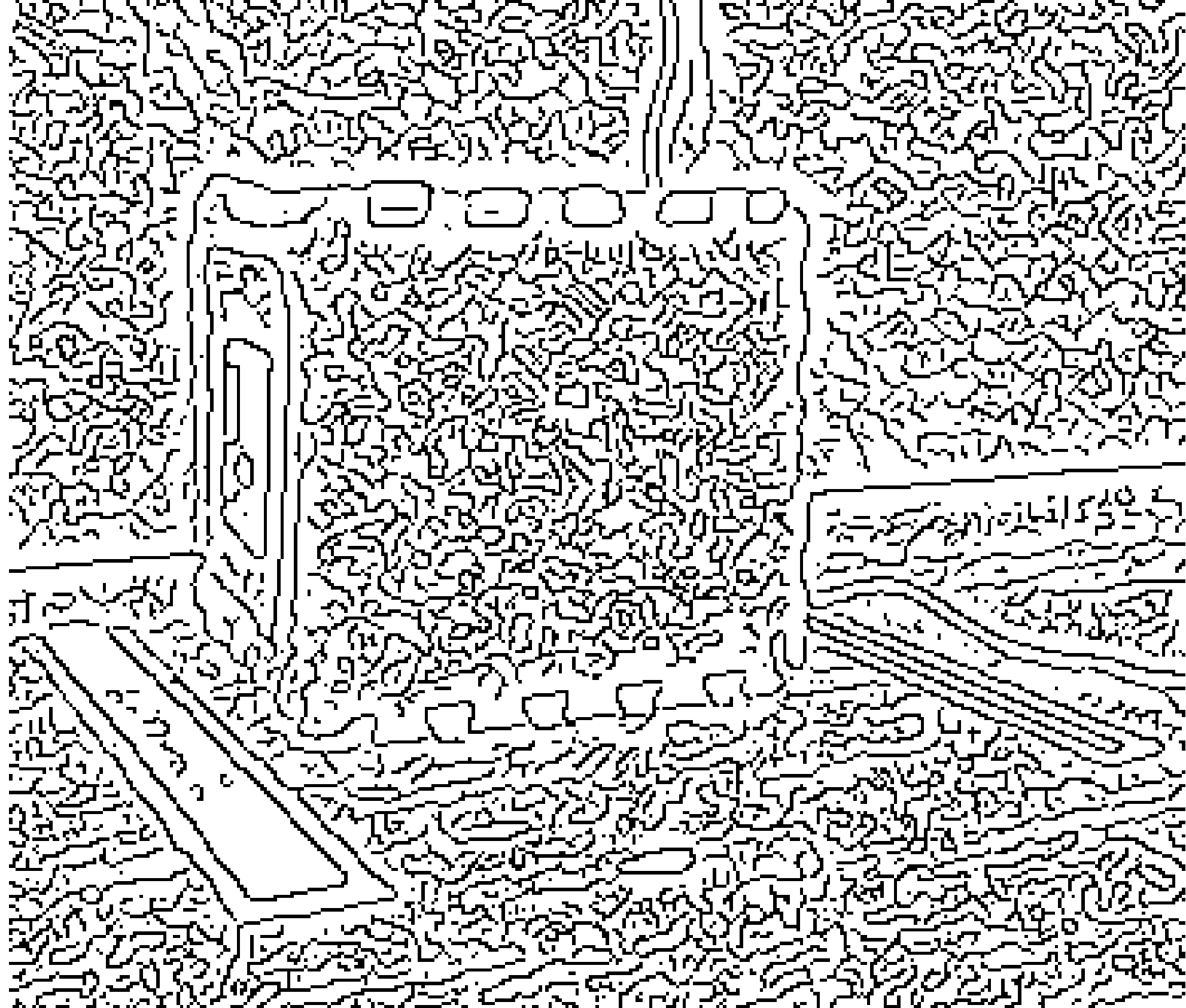


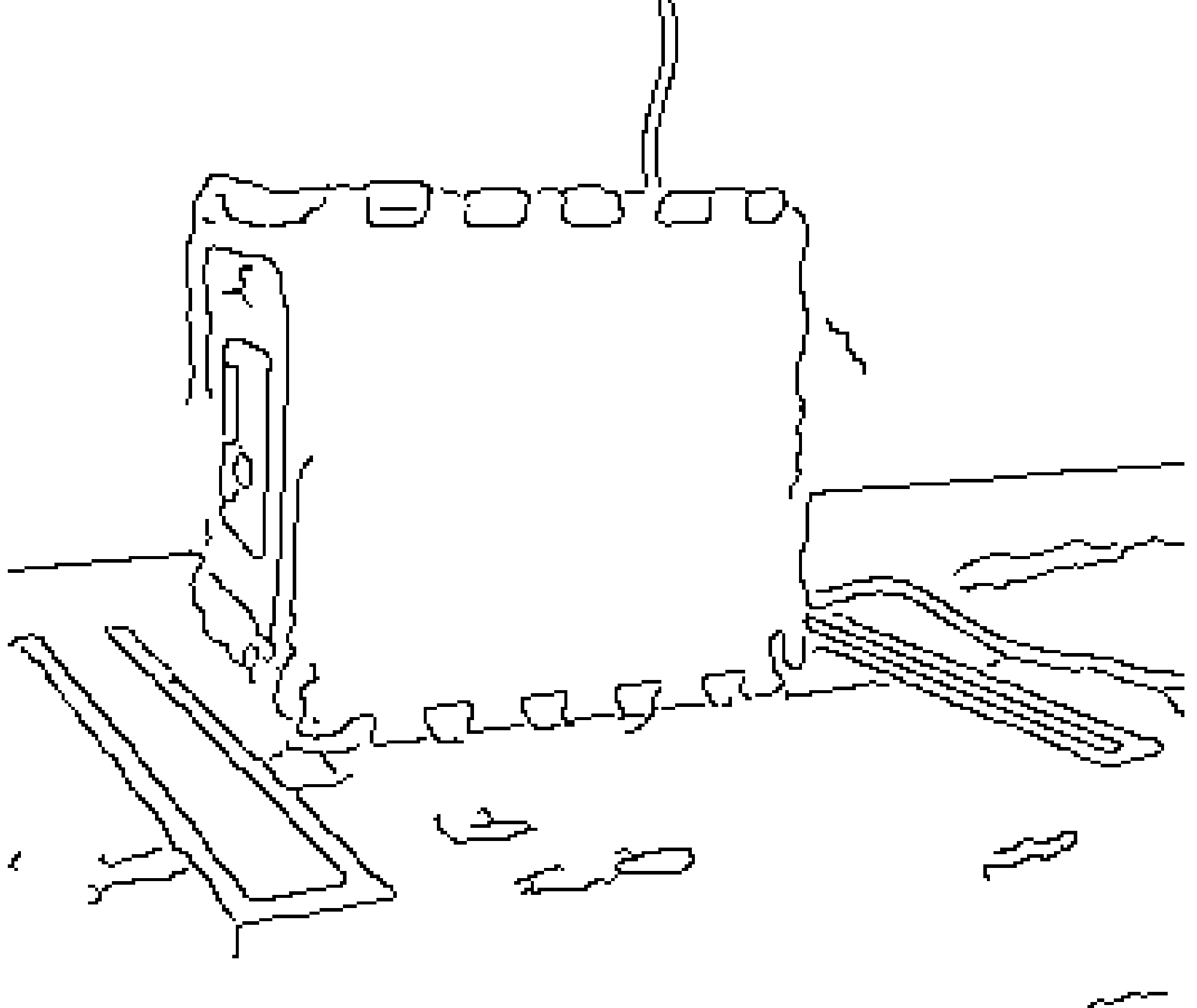


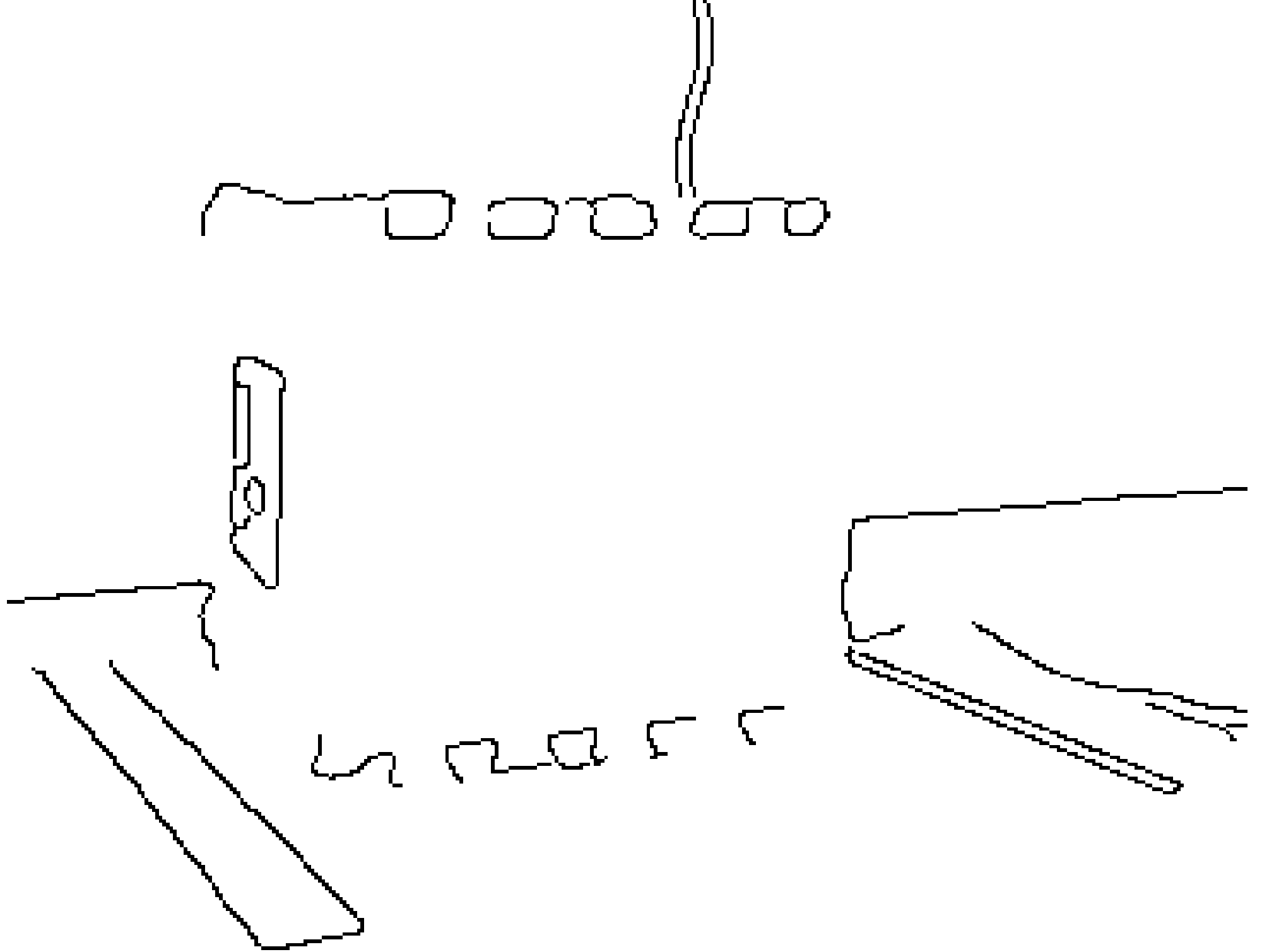


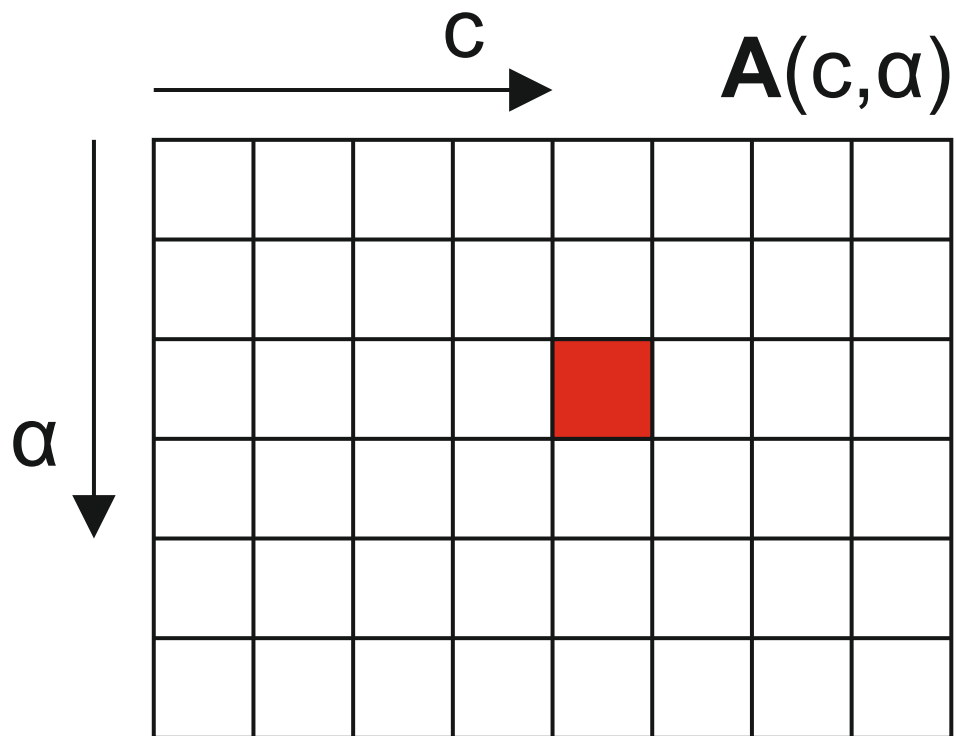
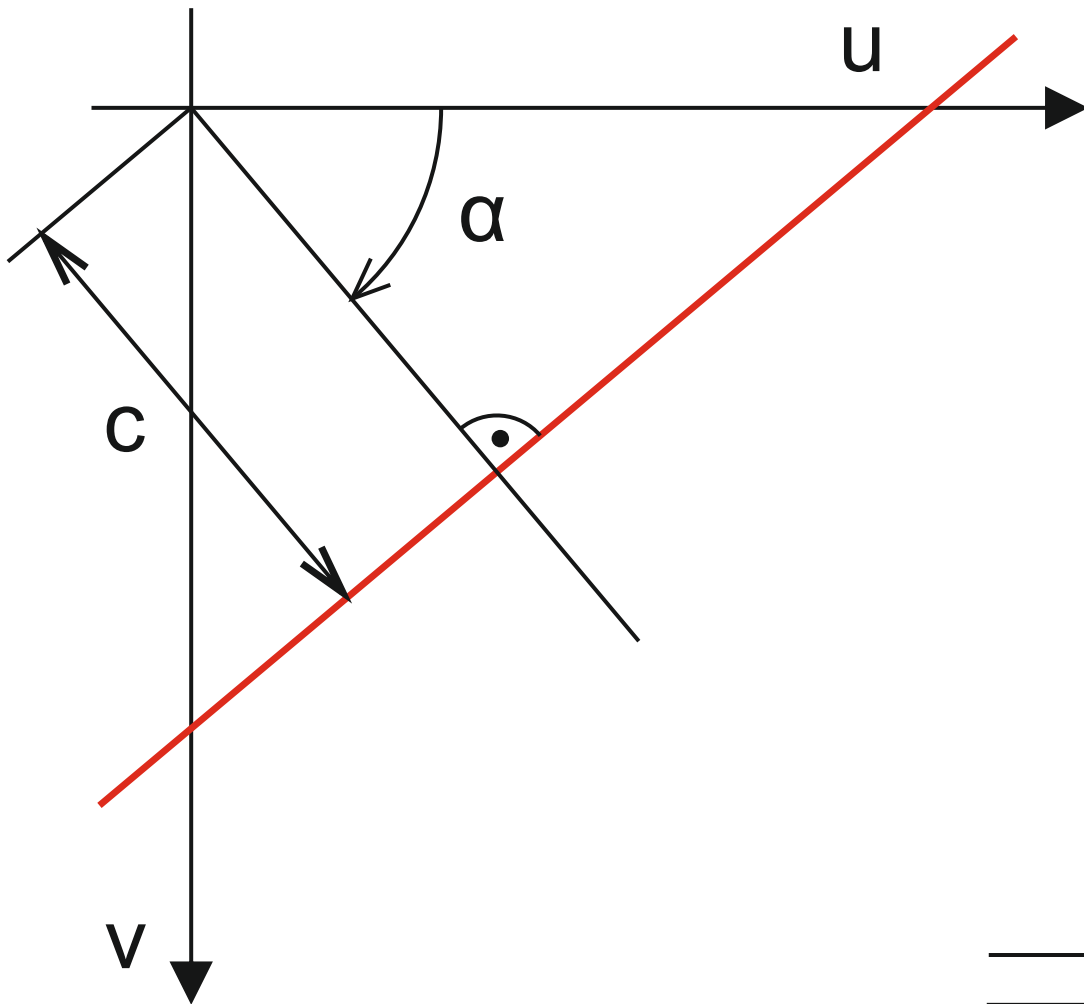


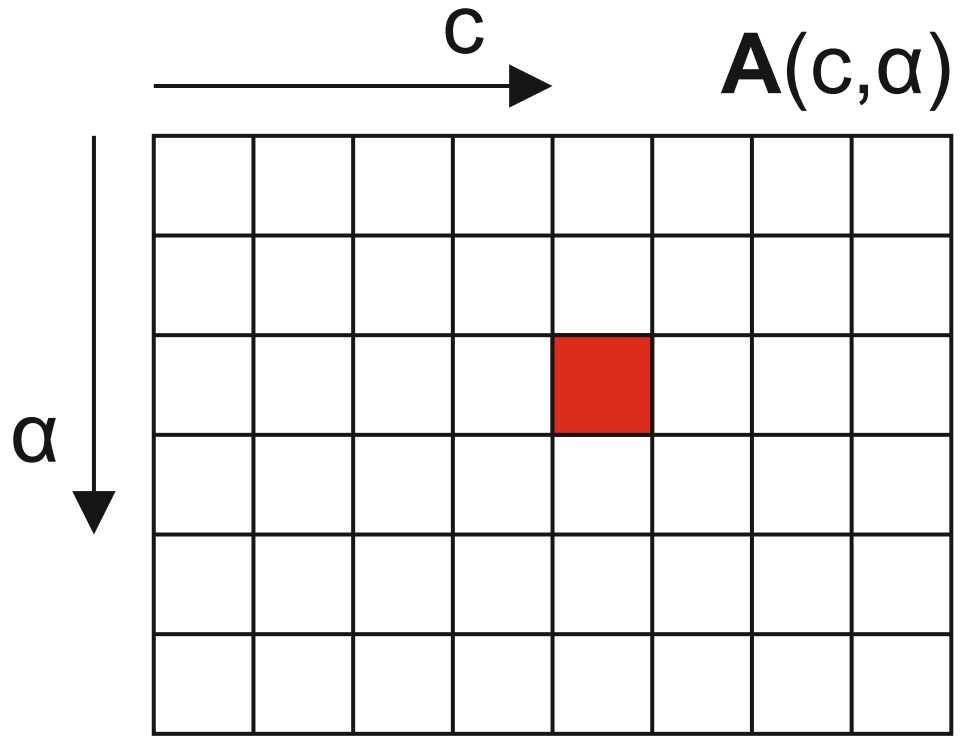
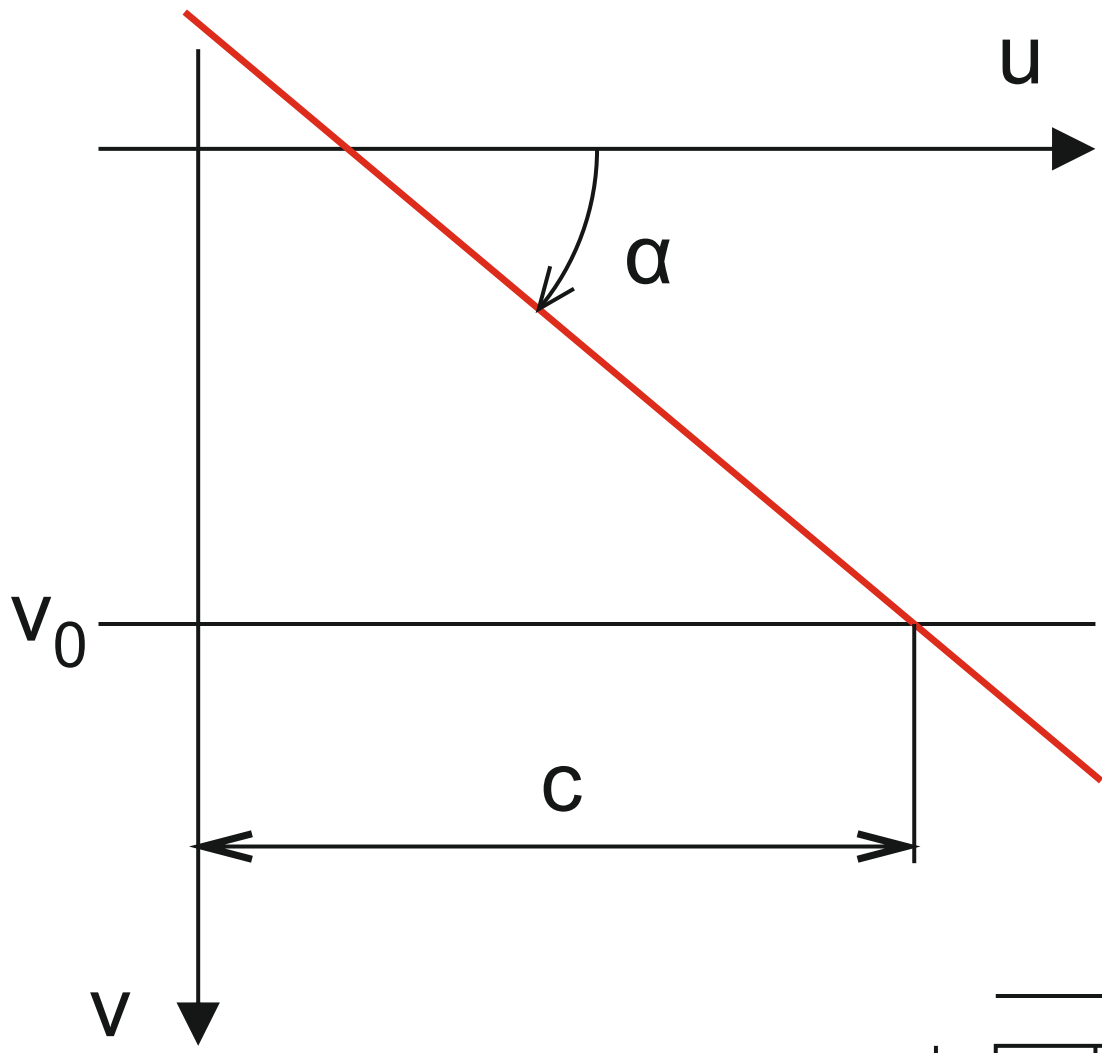




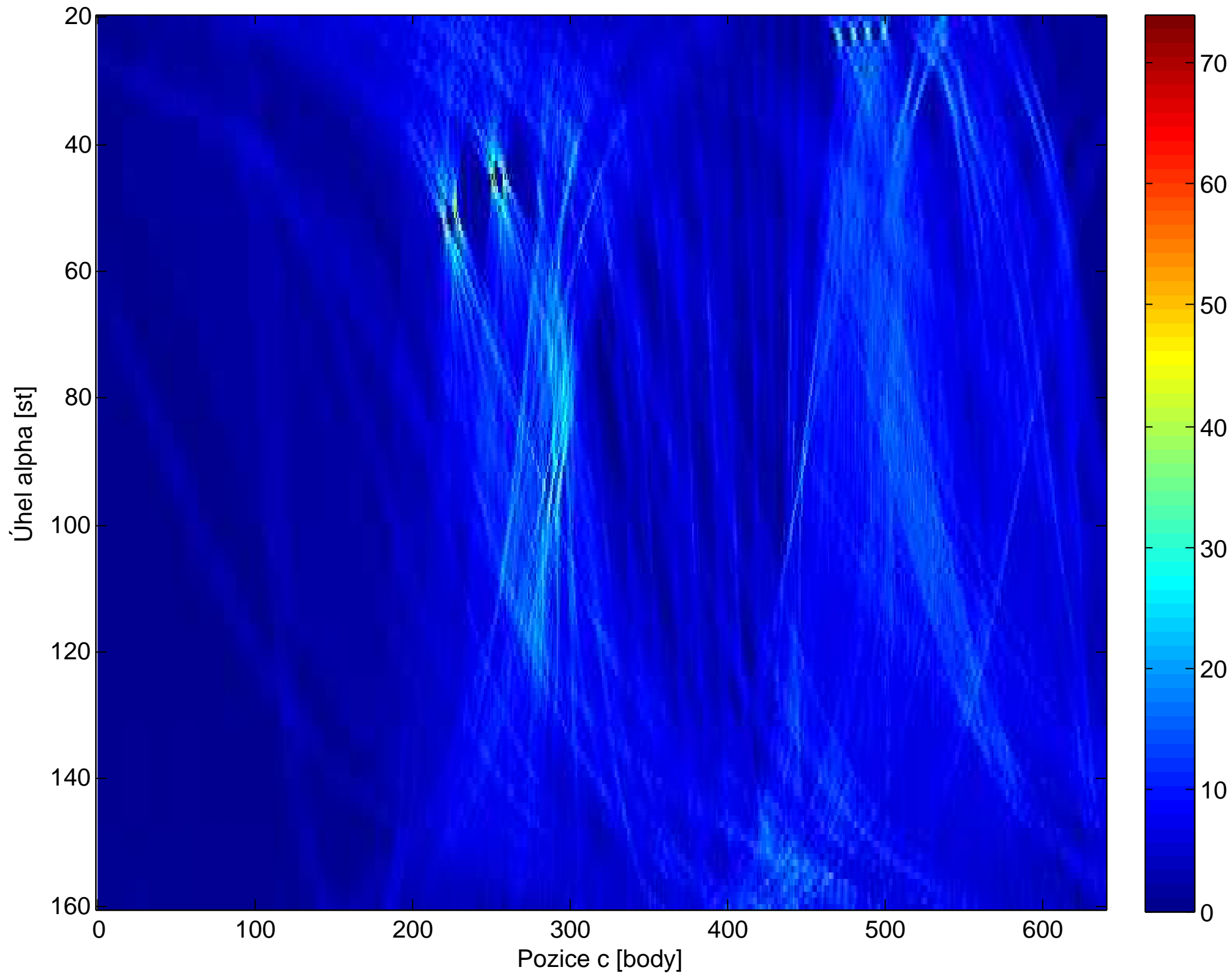


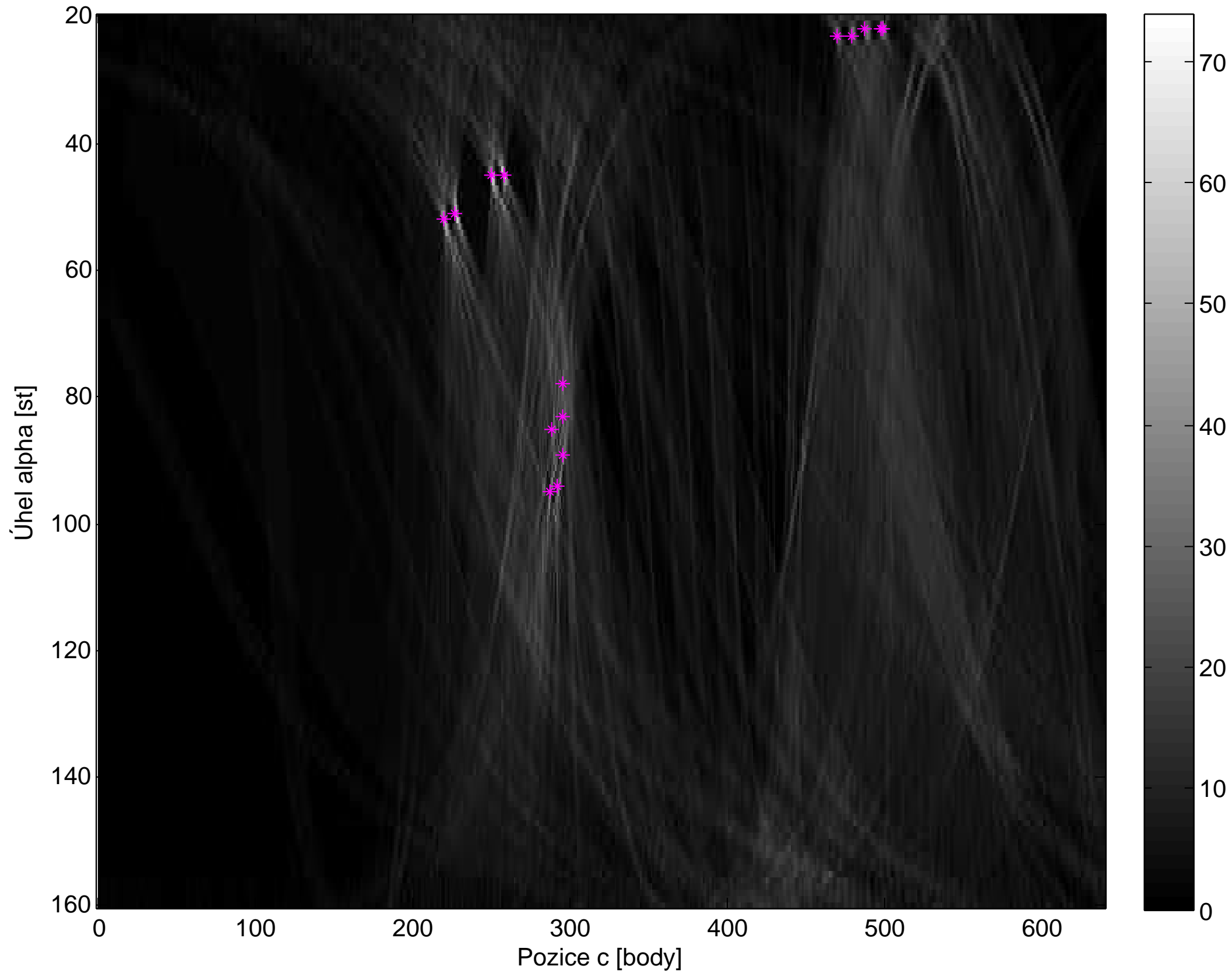




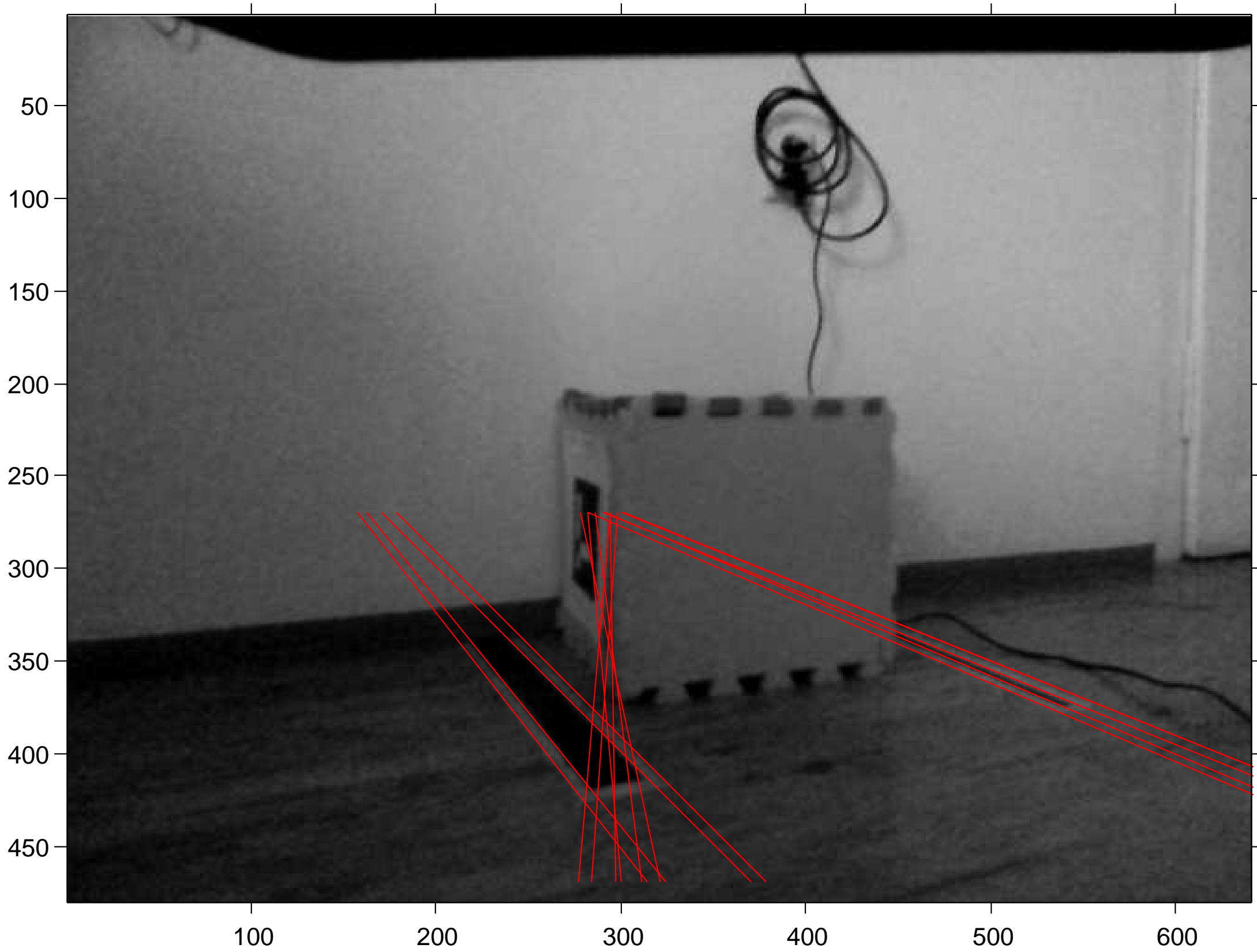














50  
100  
150  
200  
250  
300  
350  
400  
450



100

200

300

400

500

600

