Multiple View Geometry exercise 2

Hodaya Koslowsky

May 2020

1 Calibrated vs. uncalibrated reconstruction

ex1

For an uncalibrated camera, the relation between the 3D points and their projections is:

$$\lambda_{ij} x_{ij} = P_i X_j$$

For point j in camera i. If we don't know K, P can be any 3×3 matrix, so we can apply an invertible matrix T as follows:

$$\lambda_{ij} x_{ij} = \underbrace{P_i T}_{P_i'} \underbrace{T^{-1} X_j}_{X_j'}$$

Which give us another valid solution for the camera matrix and the 3D points.

computer ex1

In figure (1) we can see the 3D points and the 9 cameras. This figure makes sense when viewing the images. We can see this clearly in figure (2), where the given image points align exactly on the projected 3D points when using the matching camera matrix. In figures (3) and (4) we are giving evidence to the property we proved in ex1. To be exact, we showed that:

$$\lambda x_i = P T_k^{-1} T_k X_i$$

Are also solutions for the 3D points and camera matrices.

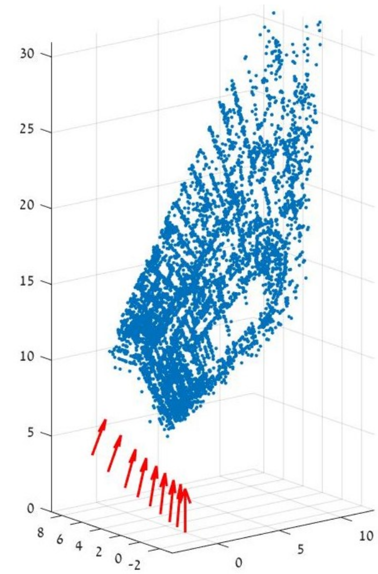

Figure 1: The 3D points and the cameras centers and principle axis

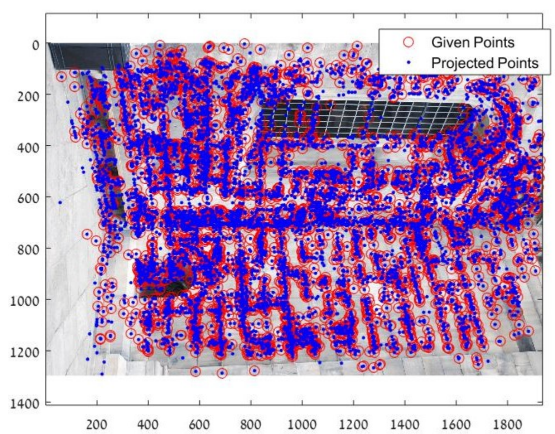

Figure 2: The image from camera 1 with the given image points and the projected points using the first camera matrix

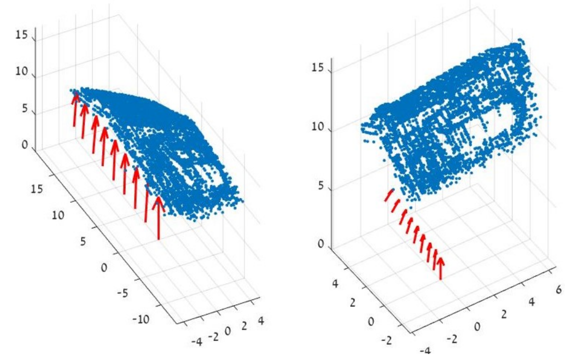

Figure 3: The 3D points and camera matrices after being transformed with T_1 (left) and T_2 (right).

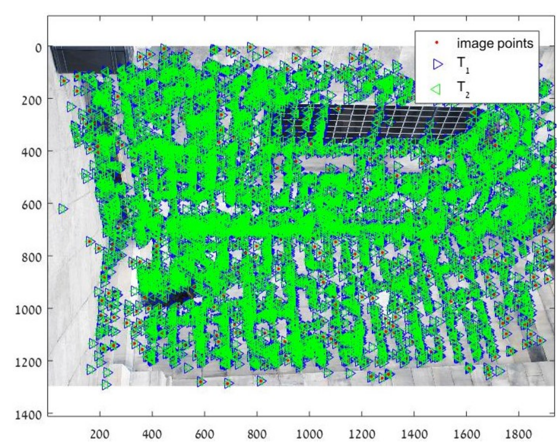

Figure 4: The image and projected points onto the first camera.

ex2

When the cameras are calibrated we know the camera matrices K_i. In this case, the system can be expressed as:

$$\lambda_{ij} x_{ij} = K_i \begin{bmatrix} R_i & t_i \end{bmatrix} X_j$$

The matrix K is invertible, so we can multiply by its inverse from the left to get:

$$\lambda_{ij} \underbrace{K_i^{-1} x_{ij}}_{\tilde{x}_{ij}} = \begin{bmatrix} R_i & t_i \end{bmatrix} X_j$$

Where $\tilde{x}_{ij}$ are called normalized points. Using this system we can see that not any 3×3 matrix will keep the new camera matrix's form of rotation and translation. In order to maintain the structure, the transformation matrix T should be of the form:

$$T = \begin{bmatrix} sQ & v \\ 0 & 1 \end{bmatrix}$$

Where $s > 0$ and Q is a rotation matrix.

$$\begin{aligned}
\lambda_{ij} \tilde{x}_{ij} &= \begin{bmatrix} R_i & t_i \end{bmatrix} \begin{bmatrix} sQ & v \\ 0 & 1 \end{bmatrix} T^{-1} X_j \\
&= \begin{bmatrix} sR_iQ & R_i v + t_i \end{bmatrix} T^{-1} X_j \\
&= s \begin{bmatrix} R_iQ & \frac{1}{s}(R_i v + t_i) \end{bmatrix} T^{-1} X_j \\
\frac{\lambda_{ij}}{s} \tilde{x}_{ij} &= \underbrace{\begin{bmatrix} R_iQ & \frac{1}{s}(R_i v + t_i) \end{bmatrix}}_{[\tilde{R}_i \; \tilde{t}_i]} \underbrace{T^{-1} X_j}_{\tilde{X}_j}
\end{aligned}$$

By using this structure for T we were able to maintain the rotation and translation form of the camera matrix, since the product of two rotation matrices is also a rotation matrix.

2 Camera Calibration

The camera matrix, K and it's inverse (assuming aspect ratio of 1 and no skew) have the following form:

$$KK^{-1} = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/f & 0 & -x_0/f \\ 0 & 1/f & -y_0/f \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We can factorize K^{-1} to two matrices:

$$K^{-1} = \underbrace{\begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}}_{B}$$

Since K transforms us from the "real world" to the image world, K^{-1} does the opposite path. It multiplies an image point $(x, y, 1)^T$. First the matrix B translates x,y to the opposite direction of the camera center, and then the matrix A scales them down by a factor of the focal length.

$$\begin{bmatrix} 1/f & 0 & -x_0/f \\ 0 & 1/f & -y_0/f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{f} \begin{bmatrix} x - x_0 \\ y - y_0 \\ f \end{bmatrix}$$

The point $(x_0, y_0, 1)^T$ is mapped to $(0, 0, 1/f)^T$ and the point $(x_0, y_0, f)^T$ is mapped to $(0, 0, 1)^T$.
Assume the following K:

$$K = \begin{bmatrix} 320 & 0 & 320 \\ 0 & 320 & 240 \\ 0 & 0 & 1 \end{bmatrix} \quad K^{-1} = \begin{bmatrix} 1/320 & 0 & -1 \\ 0 & 1/320 & -3/4 \\ 0 & 0 & 1 \end{bmatrix}$$

We can convert the points from "image world" to "real world":

$$K^{-1} \begin{bmatrix} 0 \\ 240 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

$$K^{-1} \begin{bmatrix} 640 \\ 240 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

By drawing two lines from $(0, 0, 0)^T$ to the two points we get a triangle with a base of size 2 and a line perpendicular to the base with a size of 1. This gives us an angle of 90°.
The camera center and the principle axis remain unchanged when applying K:
The camera center is calculated by finding the null space of the camera matrix. Trivially:

$$x \in \mathcal{N}([R\ t]) \rightarrow x \in \mathcal{N}(K[R\ t])$$

Moreover, since K is invertible, it has a trivial null space and:

$$x \in \mathcal{N}(K[R\ t]) \rightarrow x \in \mathcal{N}([R\ t])$$

Therefor they have the same null space, and since it represents the camera center, they have the same one.
The principle axis is calculated by the third row of the camera matrix, so we need to show that the third row is unchanged under application of K.

$$\begin{bmatrix} & -K_1- & \\ & -K_2- & \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} & -R_1- & & t_1 \\ & -R_2- & & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} = \begin{bmatrix} & & & \\ & & & \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$$

Where K_i is the i'th row of K and R_i is the i'th row of R. We can see that the third row is unchanged, and so is the principle axis.

ex4

Knowing the calibration matrix K, we can calculate K^{-1} and multiply P from the left to get the normalized P:

$$\tilde{P} = K^{-1}P = \begin{bmatrix} 1/1000 & 0 & -1/2 \\ 0 & 1/1000 & -1/2 \\ 0 & 0 & 1 \end{bmatrix} P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \end{bmatrix}$$

Given a point $x = (x_1, x_2, 1)^T$:

$$K^{-1}x = \begin{bmatrix} (x_1 - 500)/1000 \\ (x_2 - 500)/1000 \\ 1 \end{bmatrix}$$

So the points in normalized units become:

- $(0, 0, 1) \to (-0.5, -0.5, 1)$
- $(1000, 0, 1) \to (0.5, -0.5, 1)$
- $(0, 1000, 1) \to (-0.5, 0.5, 1)$
- $(1000, 1000, 1) \to (0.5, 0.5, 1)$
- $(500, 500, 1) \to (0, 0, 1)$

ex5

$$KR = \begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & f \end{bmatrix} \begin{bmatrix} -R_1^T- \\ -R_2^T- \\ -R_3^T- \end{bmatrix} = \begin{bmatrix} aR_1^T + bR_2^T + cR_3^T \\ dR_2^T + eR_3^T \\ fR_3^T \end{bmatrix}$$

We know that $\|R_3\| = 1$ and that $P(3, 1:3) = fR_3^T$, which give us:

$$1 = \|R_3\| = \left\| \frac{1}{f\sqrt{2}} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \right\| = \frac{\sqrt{2}}{f\sqrt{2}} = \frac{1}{f}$$

So:

$$f = 1, \; R_3 = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

Now denote $v = P(2, 1:3)^T$. We know that $\|R_i\| = 1$, and $R_1 \perp R_2 \perp R_3$. Then:

$$v = dR_2 + eR_3$$

$$v^T R_3 = dR_2^T R_3 + eR_3^T R_3 = e$$

And in our case:

$$e = \begin{bmatrix}-700/\sqrt{2} & 1400 & 700/\sqrt{2}\end{bmatrix} \begin{bmatrix}-1/\sqrt{2} \\ 0 \\ 1/\sqrt{2}\end{bmatrix} = 700$$

Next, we can find d:

$$d = d\|R_2\| = \|v - eR_3\| = 1400$$

And:

$$R_2 = \frac{1}{d}(v - eR_3) = \frac{1}{1400} \begin{bmatrix}0 \\ 1400 \\ 0\end{bmatrix} = \begin{bmatrix}0 \\ 1 \\ 0\end{bmatrix}$$

Next, we'll find R_1, a, b, c. Denote $u = aR_1 + bR_2 + cR_3$.

$$c = u^T R_3 = \begin{bmatrix}800/\sqrt{2} & 0 & 2400/\sqrt{2}\end{bmatrix} \begin{bmatrix}-1/\sqrt{2} \\ 0 \\ 1/\sqrt{2}\end{bmatrix} = 800$$

$$b = u^T R_2 = \begin{bmatrix}800/\sqrt{2} & 0 & 2400/\sqrt{2}\end{bmatrix} \begin{bmatrix}0 \\ 1 \\ 0\end{bmatrix} = 0$$

$$a = a\|R_1\| = \|u - cR_3\| = 1600$$

Finally:

$$R_1 = \frac{1}{a}(u - cR_3) = \frac{1}{1600} \begin{bmatrix}1600/\sqrt{2} \\ 0 \\ 1600/\sqrt{2}\end{bmatrix} = \begin{bmatrix}\frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}}\end{bmatrix}$$

In summary:

$$R = \begin{bmatrix}\frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}}\end{bmatrix} \quad K = \begin{bmatrix}1600 & 0 & 800 \\ 0 & 1400 & 700 \\ 0 & 0 & 1\end{bmatrix}$$

From K we can learn:

- $f = 1400$
- $\gamma = 8/7$
- $s = 1$
- $(x_0, y_0) = (800, 700)$

computer ex2

$$K = K_2 = \begin{bmatrix}2,394 & 0 & 932 \\ 0 & 2,398 & 628 \\ 0 & 0 & 1\end{bmatrix} K_1 = \begin{bmatrix}2,394 & 0 & 932 \\ 0 & 600 & 628 \\ 0 & 0 & 1\end{bmatrix}$$

3 Direct Linear Transformation

ex6

The problem:

$$\min_v \|Mv\|^2$$

Always has the zero solution, trivially when $v = 0$. This solution is avoided by constricting $\|v\|^2 = 1$.
Assume the singular value decomposition of $M = U\Sigma V^T$.
$M \in \mathbb{R}^{m\times n}, U \in \mathbb{R}^{m\times m}, \Sigma \in \mathbb{R}^{m\times n}, V \in \mathbb{R}^{n\times n}$ and U, V are unitary. Unitary matrices don't change the norm of a vector they multiply, so:

1. $\|Mv\|^2 = \|U\Sigma V^T v\|^2 = \|\Sigma V^T v\|^2$
2. $\|v\| = 1 \rightarrow \|V^T v\| = 1$

Denote $\tilde{v} = V^T v$. We can rephrase the problem:

$$\min_{\|\tilde{v}\|^2=1} \|\Sigma\tilde{v}\|^2$$

(In fact we only changed the basis of the vector v using the columns of V which are the eigenvectors of $M^H M$.) Thanks to the fact that U, V are unitary and we have properties (1) and (2) we have the exact same minimal value.
Assume we have the optimal $\tilde{v}$. The original optimization variable can be found by $v = VV^T v = V\tilde{v}$.
If we rephrase the problem again:

$$\min_{\|\tilde{v}\|^2=1} \sum_{i=1}^{n} \sigma_i^2 \tilde{v}_i^2$$

We can see that an optimal solution will be the vector $\tilde{v} = (0, 0, \ldots, 0, 1)^T$, which will give us a value of the minimal singular value σ_n. Notice that $-\tilde{v}$ gives the exact same solution, so we have here at least two solutions.
If $r = rank(M) < n$, then $dim\mathcal{N}(M) = n - r$, and any $v \in \mathcal{N}(M)$ gives an optimal solution of zero.

ex7

If:

$$\tilde{x} \sim Nx \text{ and } \tilde{x} \sim \tilde{P}X$$

Then:

$$Nx \sim \tilde{P}X$$

Assuming a normalization matrix is invertible:

$$x \sim N^{-1}\tilde{P}X$$

So we can express the new camera matrix as $P = N^{-1}\tilde{P}$.

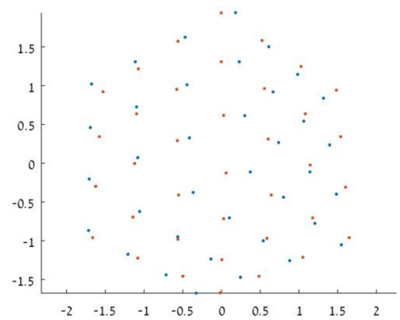

Figure 5: Projected points after normalization to zero mean and one variance. Blue-camera 1, red-camera 2.

computer ex3

After normalizing the projected points as in figure (5), we construct two matrices (M) and solve DLT.
For the first camera, we got that the smallest eigenvalue of M is 0.015, and for the second camera 0.012. These values are equal $\|Mv\|$, where v is the last column of the matrix V that is returned from applying SVD on M. When projecting the

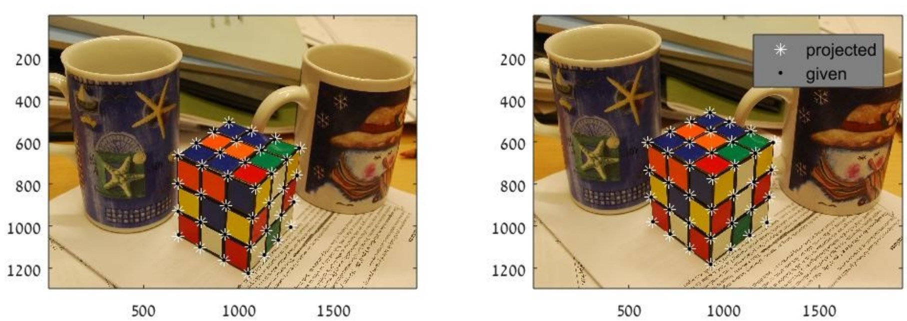

Figure 6: The projected points from the 3D points, together with the given points on the image. They match exactly.

calculated points and plotting them on the images with the given image points, they align beautifully, as seen in figure (6). In figure (7) we can view the cameras location and viewing angle relative to the 3D points. These figures make sense when considering the images. When factoring the first camera matrix we get

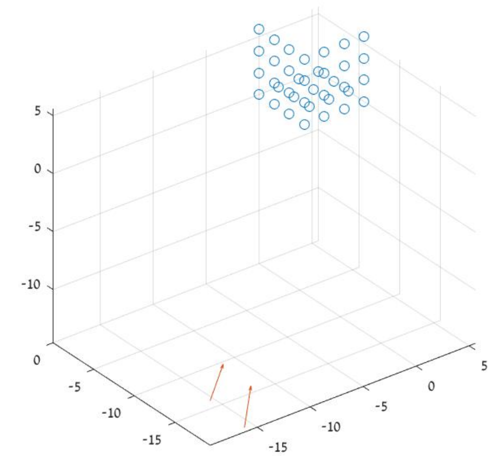

Figure 7: The 3D points and the two cameras, computed from the camera matrices.

the following calibration matrix:

$$K = \begin{bmatrix} 2448 & -18 & 960 \\ 0 & 2446 & 675 \\ 0 & 0 & 1 \end{bmatrix}$$

Since we have the original 3D points we can't have ambiguity. As opposed to question 1, where we did not know them and could multiply both P and X by some transformations.

RMS

The root mean square error of the projected and given points of camera 1 is 2.37 and 2.73 for the x and y coordinates respectively. Without normalization we get RMS of 3.3 and 3.62.
When taking only 7 of the points the differences are stronger:
3.09 and 2.69 when normalizing and 9.46 and 8.54 when not normalizing.
We can conclude that when having few data points normalization becomes more important when trying to avoid mistakes in the DLT calculations.

Computer ex4

Some of the matches in figure (8) seem great - the ones that create horizontal lines. However, there are several outliers.

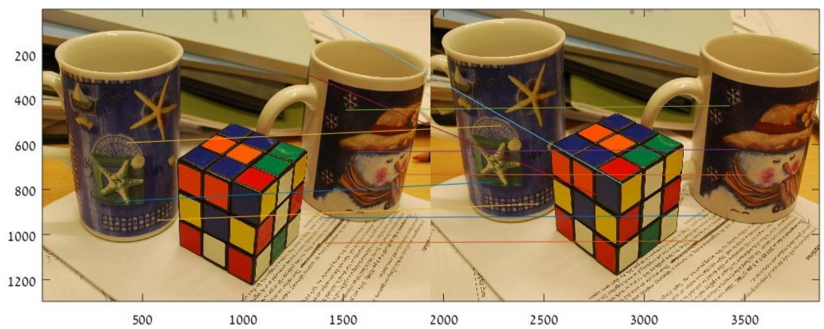

Figure 8: SIFT example matches

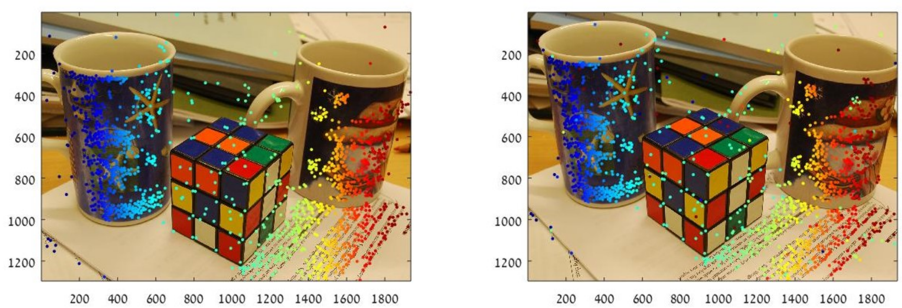

Figure 9: All SIFT matches

Computer ex5

Finally, in figure (11) we see the camera looking at the 3D scene - with the given model points of the cube and the points we calculated with triangulation. We can see the table and the glass nicely align with the images.

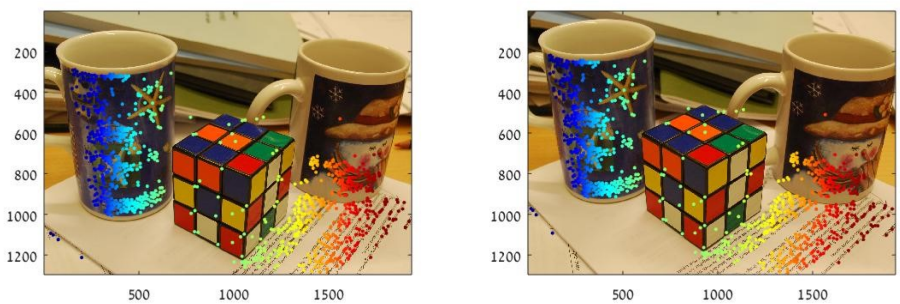

Figure 10: SIFT good matches

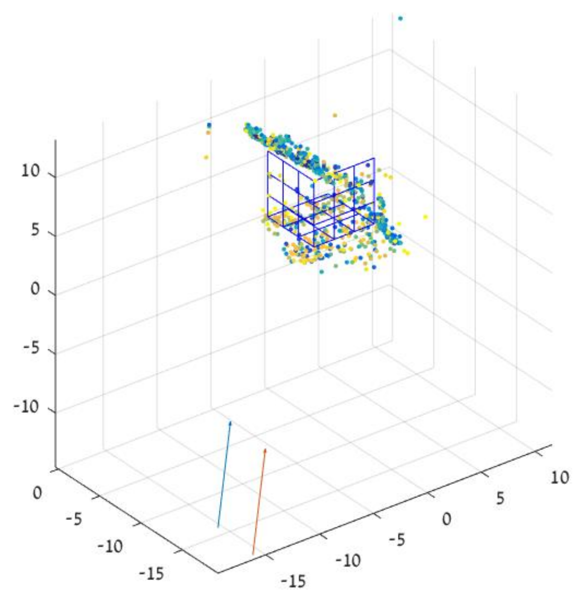

Figure 11: 3D points from triangulation and the model points of the cube.