

A Closed-Form Solution for the Uniform Sampling of the Epipolar Line via Non-Uniform Depth Sampling

Engin Tola
Computer Vision Laboratory
École Polytechnique Fédérale de Lausanne
Switzerland

engin.tola@epfl.ch
<http://cvlab.epfl.ch/~tola/>

Technical Report EPFL/CVLAB - 2010

Given two camera calibrations, this report presents a closed form algorithm that computes a sequence of 3D points such that they all project to a single location on one camera and that their projection forms a uniformly sampled line on the other camera.

Let a camera be parametrized by its intrinsic parameters \mathbf{K} , and extrinsic parameters of rotation matrix \mathbf{R} and camera center \mathbf{C} . The projection of a 3D point \mathbf{X} is defined as

$$\lambda \mathbf{x} = \mathbf{K}\mathbf{R}(\mathbf{X} - \mathbf{C}) \quad (1)$$

with $\mathbf{x} = [x \ y \ 1]^T$ its image coordinates and λ its depth. Then, the back-projected ray emanating from point $\mathbf{x} = [x \ y \ 1]^T$ is parametrized in terms of the depth variable λ as:

$$\mathbf{X}(\lambda) = \lambda \mathbf{R}^T \mathbf{K}^{-1} \mathbf{x} + \mathbf{C} \quad (2)$$

Given two camera calibration matrices $P_0 = (\mathbf{K}_0, \mathbf{R}_0, \mathbf{C}_0)$ and $P_1 = (\mathbf{K}_1, \mathbf{R}_1, \mathbf{C}_1)$, we would like to sample the back-projected line $\mathbf{X}(\lambda)$ so that the projected samples on camera P_1 are uniformly separated (see Figure 1).

The two points separated by depth $d\lambda$ on $\mathbf{X}(\lambda)$ is equal to:

$$\mathbf{X}(\lambda) = \lambda \mathbf{R}_0^T \mathbf{K}_0^{-1} \mathbf{x} + \mathbf{C}_0 \quad (3)$$

$$\mathbf{X}(\lambda + d\lambda) = (\lambda + d\lambda) \mathbf{R}_0^T \mathbf{K}_0^{-1} \mathbf{x} + \mathbf{C}_0 \quad (4)$$

Then, the projection of these points on image P_1 are

$$\omega \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \lambda \mathbf{K}_1 \mathbf{R}_1 \mathbf{R}_0^T \mathbf{K}_0^{-1} \mathbf{x} + \mathbf{K}_1 \mathbf{R}_1 (\mathbf{C}_0 - \mathbf{C}_1) \quad (5)$$

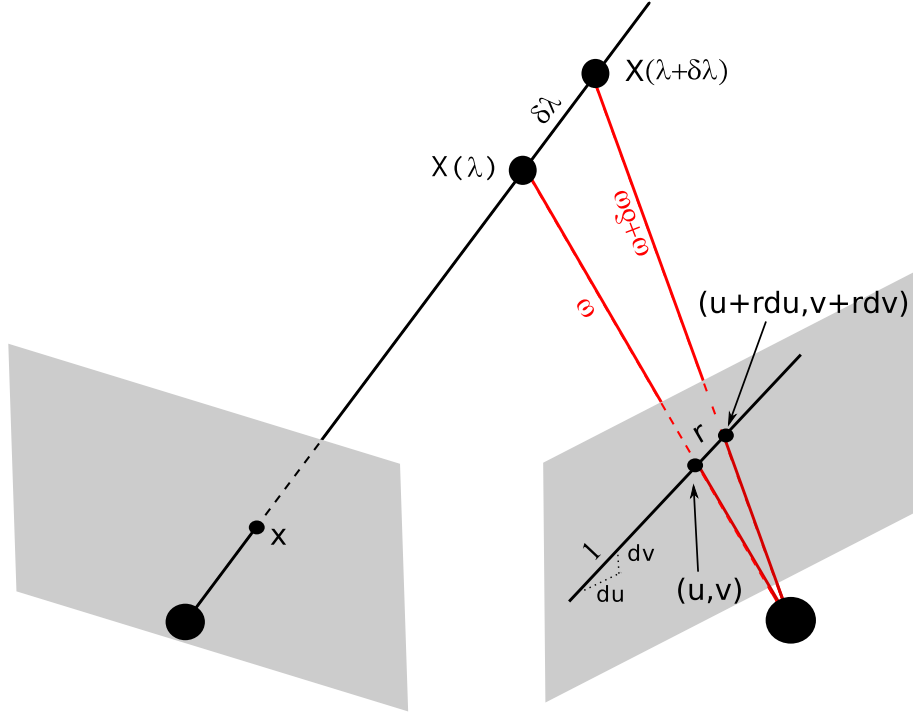


Figure 1: **Uniform Step Sampling Framework**

and

$$(\omega + d\omega) \begin{bmatrix} u + rdu \\ v + rdv \\ 1 \end{bmatrix} = (\lambda + d\lambda) \mathbf{K}_1 \mathbf{R}_1 \mathbf{R}_0^T \mathbf{K}_0^{-1} \mathbf{x} + \mathbf{K}_1 \mathbf{R}_1 (\mathbf{C}_0 - \mathbf{C}_1) \quad (6)$$

with (u, v) and ω the coordinates and depth of $\mathbf{X}(\lambda)$ on image P_1 respectively, $w + d\omega$ the depth of the point $\mathbf{X}(\lambda + d\lambda)$, (du, dv) the slope of the epipolar line and r is the sampling resolution.

The slope of the epipolar line can be found by projecting any two points $\mathbf{X}(\lambda_{min})$ and $\mathbf{X}(\lambda_{max})$ onto P_1 . Let the coordinates of the projections are (u_{min}, v_{min}) and (u_{max}, v_{max}) . The slope is then computed as:

$$dl = \sqrt{(u_{max} - u_{min})^2 + (v_{max} - v_{min})^2}$$

$$du = (u_{max} - u_{min})/dl \quad (7)$$

$$dv = (v_{max} - v_{min})/dl .$$

Let

$$\begin{aligned} \mathbf{a} &= \mathbf{K}_1 \mathbf{R}_1 \mathbf{R}_0^T \mathbf{K}_0^{-1} \mathbf{x} \\ \mathbf{b} &= \mathbf{K}_1 \mathbf{R}_1 (\mathbf{C}_0 - \mathbf{C}_1) \end{aligned} \quad (8)$$

and rewrite Equations 5 and 6 as

$$\omega \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \lambda \mathbf{a} + \mathbf{b} \quad (9)$$

and

$$(\omega + d\omega) \begin{bmatrix} u + rdu \\ v + rdv \\ 1 \end{bmatrix} = (\lambda + d\lambda) \mathbf{a} + \mathbf{b} . \quad (10)$$

Rearranging Equations 9 and 10, one can see the relationship between the changes in the depths for two views as:

$$\mathbf{a}d\lambda = \omega \begin{bmatrix} rdu \\ rdv \\ 0 \end{bmatrix} + d\omega \begin{bmatrix} u + rdu \\ v + rdv \\ 1 \end{bmatrix} \quad (11)$$

If we denote the i^{th} element of vector \mathbf{a} with a_i and expand the third row of the Equation 11,

$$a_2d\lambda = d\omega , \quad (12)$$

the first two rows of Equation 11, after some rearranging, becomes

$$\begin{aligned} d\lambda &= \frac{\omega rdu}{a_0 - a_2(u + rdu)} \\ d\lambda &= \frac{\omega rdv}{a_1 - a_2(v + rdv)} . \end{aligned} \quad (13)$$

Equation 13 is the update in the depth with respect to camera P_0 that one must make in order to move r pixels in the direction of (du, dv) on camera P_1 where (u, v) and ω is the current projection and current depth with respect to camera P_1 . (see Figure 1). Thus, the sequence of 3D points that will have a uniform sampling projection can be computed by iterating this process.

The pseudo-code that computes the sampling locations and the depths associated with these points are given in Algorithm 1. A Matlab implementation for the algorithm is given in [Tola, 2010].

References

[Tola, 2010] Tola, E. (2010). Uniform sampling of epipolar line via non-uniform depth sampling - http://cvlab.epfl.ch/~tola/open_source.html#episample.

Algorithm 1: Non-uniform Space Sampling

Require: Camera Parameters: $(\mathbf{K}_0, \mathbf{R}_0, \mathbf{C}_0)$ and $(\mathbf{K}_1, \mathbf{R}_1, \mathbf{C}_1)$

Require: Point location \mathbf{x}

Require: Depth range $(\lambda_{min}, \lambda_{dmax})$

Require: Sampling resolution r

- 1: $\mathbf{a} \leftarrow \mathbf{K}_1 \mathbf{R}_1 \mathbf{R}_0^T \mathbf{K}_0^{-1} \mathbf{x}$
 - 2: $\mathbf{b} \leftarrow \mathbf{K}_1 \mathbf{R}_1 (\mathbf{C}_0 - \mathbf{C}_1)$
 - 3: $u_{min} \leftarrow \frac{\lambda_{min} a_0 + b_0}{\lambda_{min} a_2 + b_2}, \quad v_{min} \leftarrow \frac{\lambda_{min} a_1 + b_1}{\lambda_{min} a_2 + b_2}$
 - 4: $u_{max} \leftarrow \frac{\lambda_{max} a_0 + b_0}{\lambda_{max} a_2 + b_2}, \quad v_{max} \leftarrow \frac{\lambda_{max} a_1 + b_1}{\lambda_{max} a_2 + b_2}$
 - 5: $dl \leftarrow \sqrt{(u_{max} - u_{min})^2 + (v_{max} - v_{min})^2}$
 - 6: $du \leftarrow (u_{max} - u_{min})/dl, \quad dv \leftarrow (v_{max} - v_{min})/dl$
 - 7: $\omega_{min} \leftarrow \lambda_{min} a_2 + b_2$

 - 8: $\omega \leftarrow \omega_{min}$
 - 9: $\lambda \leftarrow \lambda_{min}$
 - 10: $(u, v) \leftarrow (u_{min}, v_{min})$
 - 11: **while** $\lambda < \lambda_{max}$ **do**
 - 12: **if** $|du|$ not equal to 0 **then**
 - 13: $d\lambda \leftarrow \frac{\omega r du}{a_0 - a_2(u + r du)}$
 - 14: **else**
 - 15: $d\lambda \leftarrow \frac{\omega r dv}{a_1 - a_2(v + r dv)}$
 - 16: **end if**
 - 17: $\lambda \leftarrow \lambda + d\lambda$
 - 18: $(u, v) \leftarrow (u + r du, v + r dv)$
 - 19: $X \leftarrow \lambda \mathbf{R}^T \mathbf{K}^{-1} \mathbf{x} + \mathbf{C}$
 - 20: $\omega \leftarrow \lambda a_2 + b_2$
 - 21: **end while**
-