A survey on Camera Models and Affine Invariance

Stefano Melacci
DIISM - University of Siena

August 9, 2014

1 Camera models

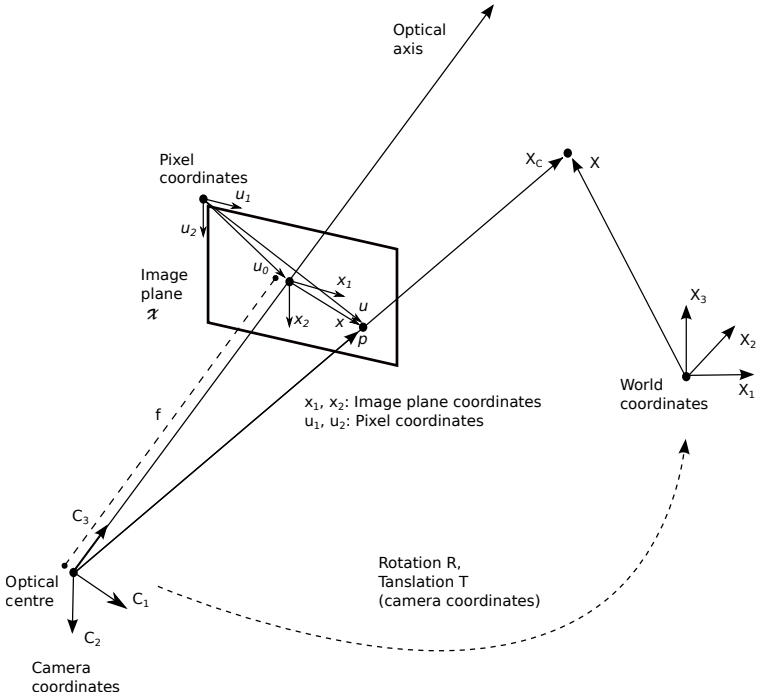

Figure 1: Camera model: perspective projection.

Before starting our analysis, we introduce the basic properties of projections that we will use throughout the report. In particular, we will introduce a generic camera model and then we will move towards a simplified model that is more suitable for the proposed study.

Figure 1 describes a simple camera model in which a perspective projection is applied to map the point X, expressed in world coordinates, onto the point u, expressed in pixel coordinates. We are given a world reference system $\perp_W = \{X_1, X_2, X_3\}$ that represents the 3D reference frame of each real-world object. Then we have a camera-based reference system $\perp_C = \{C_1, C_2, C_3\}$ that is centered on the optical centre of the camera, and an image plane frame $\perp_{\mathcal{X}} = \{x_1, x_2\}$. Finally, we consider the pixel coordinates $\perp_P = \{u_1, u_2\}$, that are centered in the upper left corner of the CCD.

The projection model considered here is inspired by the pin-hole camera and it is based on perspective projection. In particular, the rigid body motion between the world-frame and the camera-frame consists of a rotation of the points in $\perp_W$ about X_3, X_2, and X_1 of the angles φ_3, φ_2, φ_1, respectively, followed by a translation $T_C = [T_{C_1}, T_{C_2}, T_{C_3}]'$ (where T_C is the center of the word-frame expressed in camera coordinates). For convenience, the rotation is described by a matrix $R \in I\!R^{3,3}$,

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\varphi_1 & -sin\varphi_1 \\ 0 & sin\varphi_1 & cos\varphi_1 \end{bmatrix} \cdot \begin{bmatrix} cos\varphi_2 & 0 & sin\varphi_2 \\ 0 & 1 & 0 \\ -sin\varphi_2 & 0 & cos\varphi_2 \end{bmatrix} \cdot \begin{bmatrix} cos\varphi_3 & -sin\varphi_3 & 0 \\ sin\varphi_3 & cos\varphi_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$X_C = RX + T_C. \tag{1}$$

The perspective projection is defined by the focal length f, that is the distance of the camera-frame origin and the image plane $\mathcal{X}$. The C_3 coordinate of $\perp_C$ is assumed to be orthogonal to the image plane $\mathcal{X}$. The perspective projection maps a point expressed in $\perp_C$ onto a point of the image plane, expressed in $\perp_{\mathcal{X}}$. Formally,

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{f}{X_{C_3}} \cdot \begin{bmatrix} X_{C_1} \\ X_{C_2} \end{bmatrix}. \tag{2}$$

The intersection of the optical axis with $\mathcal{X}$ is called *principal point*, and its coordinates in $\perp_P$ are given by $u_0 = [u_{0_1}, u_{0_2}]'$, whereas they are $[u_0, f]$ in $\perp_C$. We indicate with k_{u_1} and k_{u_2} the ratio between the coordinates in $\perp_P$ and the ones in $\perp_{\mathcal{X}}$, whereas s represents the skew between the axis of $\perp_P$. We can pass from image plane coordinates to pixel coordinates with

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} k_{u_1} x_1 + s x_2 \\ k_{u_2} x_2 \end{bmatrix} + u_0. \tag{3}$$

From now on we indicate with $\tilde{O}$ the homogeneous coordinates of a generic point O, i.e. if $O = [O_1, O_2, O_3]'$ then $\tilde{O} = [\lambda O_1, \lambda O_2, \lambda O_3, \lambda]'$. The camera model can be expressed as a combination of rigid-body transform, perspective projection, and CCD imaging, and represented as

$$\tilde{u} = \begin{bmatrix} k_{u_1} & s & u_{0_1} \\ 0 & k_{u_2} & u_{0_2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \left[\begin{array}{ccc|c} & R & & T_C \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tilde{X} \tag{4}$$

or, more compactly,

$$\tilde{u} = K[R|T_C] \cdot \tilde{X} := \begin{bmatrix} \alpha_{u_1} & \alpha_s & u_{0_1} \\ 0 & \alpha_{u_2} & u_{0_2} \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} & & & \\ & R & & T \\ & & & \end{array}\right] \tilde{X} \tag{5}$$

where $\alpha_{u_1} = f k_{u_1}$, $\alpha_{u_2} = f k_{u_2}$ and $\alpha_s = f s$. The matrix K is commonly referred to as the *calibration matrix*, and it only depends on the intrinsic camera parameters. To complete the camera model we should also take into account the lens distortion effect (fisheye, radial distortion, ...), shortly resumed by a non-linear function $d(\cdot)$,

$$\tilde{u} = K \cdot d(\ [R|T_C] \cdot \tilde{X}\ ). \tag{6}$$

If we restrict our analysis to a small region of the image plane, the effects of $d(\cdot)$ are negligible. For simplicity, in the following analysis we discard the lens distortion $d(\cdot)$.

Perspective cameras belong to a larger class of cameras that are commonly referred to as *projective cameras*. Projective cameras are described by a general 3×4 matrix P_{proj},

$$\tilde{u} = P_{proj}\tilde{X} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \tilde{X} \tag{7}$$

that has 11 degrees of freedom, since the overall scaler of P does not matter (usually $p_{34} = 1$).

1.1 Affine camera

Under some specific conditions, we can simplify the perspective camera model to reduce the degrees of freedom of the map, making it more tractable. From (5) we recall that

$$\tilde{u} = K[R|T_C] \cdot \tilde{X} = \begin{bmatrix} \alpha_{u_1} & \alpha_s & u_{0_1} \\ 0 & \alpha_{u_2} & u_{0_2} \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & T_{C_1} \\ r_{21} & r_{22} & r_{23} & T_{C_2} \\ r_{31} & r_{32} & r_{33} & T_{C_3} \end{array}\right] \tilde{X} \tag{8}$$

Now, we start moving the camera far away from the scene, along C_3, and we also increase f by a positive factor to get a magnification effect (zooming) and without changing the image size. At time $t = 0$ we have $z(0) = T_{C_3}$, whereas at each time interval t equation (8) becomes

$$\tilde{u} = \begin{bmatrix} \frac{z(t)}{z(0)} \cdot \alpha_{u_1} & \frac{z(t)}{z(0)} \cdot \alpha_s & u_{0_1} \\ 0 & \frac{z(t)}{z(0)} \cdot \alpha_{u_2} & u_{0_2} \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & T_{C_1} \\ r_{21} & r_{22} & r_{23} & T_{C_2} \\ r_{31} & r_{32} & r_{33} & z(t) \end{array}\right] \tilde{X} \tag{9}$$

$$= \frac{z(t)}{z(0)} \cdot K \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & T_{C_1} \\ r_{21} & r_{22} & r_{23} & T_{C_2} \\ \frac{z(0)}{z(t)} \cdot r_{31} & \frac{z(0)}{z(t)} \cdot r_{32} & \frac{z(0)}{z(t)} \cdot r_{33} & z(0) \end{array}\right] \tilde{X} \tag{10}$$

When $t \to \infty$, we get

$$\tilde{u} = K \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & T_{C_1} \\ r_{21} & r_{22} & r_{23} & T_{C_2} \\ 0 & 0 & 0 & T_{C_3} \end{array}\right] \tilde{X} \tag{11}$$

that is an instance of *affine camera*, with the following properties and differences from the perspective model:

- the perspective camera does not preserve parallelism, length, and angle. It preserves collinearity and incidence;
- the affine camera does not preserve length and angle. It preserves collinearily incidence, and *parallelism*.

Affine cameras are also referred to as *cameras at infinity*.

Proposition 1.1 *The affine camera model is not sensitive to changes in depth along the optical axis.*

Proof: We consider the plane through the world origin perpendicular to the optical axis, and we assume, without any lack of generality, that $R = I$, so that we have $X_3 = 0$. Suppose that we move the plane by Δ along the optical axis, so that $X_3 = \Delta$. The image of any point on such plane is, in the case of a perspective camera,

$$\tilde{u}_{persp} = K \begin{bmatrix} x_1 \\ x_2 \\ T_{C_3} + \Delta \end{bmatrix} \tag{12}$$

For affine cameras we have

$$\tilde{u}_{aff} = K \begin{bmatrix} x_1 \\ x_2 \\ T_{C_3} \end{bmatrix} \tag{13}$$

so that we can see that the affine mapping is not affected by the variation of depth.

□

Proposition 1.2 *In an affine camera model, given the depth variations Δ of the objects in the scene and the average depth $T_{C_3}^{avg}$, we have*

$$u_{aff} - u_{persp} = \frac{\Delta}{T_{C_3}^{avg}}(u_{persp} - u_0) \tag{14}$$

where u_{aff} and u_{persp} are the coordinates of and an affine and a perspective projection, respectively.

Proof: Once we dehomogenize the coordinates of (12) and (13) and we substitute T_{C_3} with the average depth of the objects in the scene T_3^{avg}, it is easy to show that the proposition holds true.

□

Proposition 1.3 *When the variations of depths Δ of the objects in the scene is small compared to the average depth T_3^{avg}, and if consider that the distance of each point from the optical axis is small (i.e. small field of view) there are almost no differences between an affine or a perspective camera.*

Proof: Straightforward from Proposition (1.2).

□

Affine projections constitute a large class of projections that can be further divided inro several specific instances (weak perspective, orthogonal projection, ...). They are all characterized by the following form,

$$\tilde{u} = P_{aff}\tilde{X} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & p_{34} \end{bmatrix} \tilde{X} \tag{15}$$

that has 8 degrees of freedom since the overall scale does not matter (i.e. we can set $p_{34} = 1$).

1.1.1 Decompositions of the affine camera matrix

We can go one step further and avoid including the coordinates of the principal point, since it is not an intrinsic property of the affine camera and it depends by the particular choice of world coordinates.

Proposition 1.4 *The principal point u is not an intrinsic property of an affine camera.*

Proof: Following (11) and the result of Proposition (1.1) the affine camera projection, and, in particular, the matrix P_{aff}, can be written as[1]

$$\begin{aligned}
\tilde{u} &= P_{aff}\tilde{X} \\
&= K \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & T_{C_1} \\ r_{21} & r_{22} & r_{23} & T_{C_2} \\ 0 & 0 & 0 & 1 \end{array}\right] \tilde{X} \\
&= \left[\begin{array}{ccc} \alpha_1 & \alpha_s & u_{0_1} \\ 0 & \alpha_2 & u_{0_2} \\ 0 & 0 & 1 \end{array}\right] \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & T_{C_1} \\ r_{21} & r_{22} & r_{23} & T_{C_2} \\ 0 & 0 & 0 & 1 \end{array}\right] \tilde{X} \\
&= \left[\begin{array}{cc} K_{1:2,1:2} & u_0 \\ 0' & 1 \end{array}\right] \left[\begin{array}{cc} R_{1:2,1:3} & T_{C_{1:2}} \\ 0' & 1 \end{array}\right] \tilde{X} \\
&= \left[\begin{array}{cc} K_{1:2,1:2}R_{1:2,1:3} & K_{1:2,1:2}T_{C_{1:2}} + u_0 \\ 0' & 1 \end{array}\right] \\
&= \left[\begin{array}{cc} K_{1:2,1:2} & 0 \\ 0' & 1 \end{array}\right] \left[\begin{array}{cc} R_{1:2,1:3} & T_{C_{1:2}} + K^{-1}_{1:2,1:2}u_0 \\ 0' & 1 \end{array}\right] \tilde{X} \\
&= \left[\begin{array}{ccc} \alpha_1 & \alpha_s & 0 \\ 0 & \alpha_2 & 0 \\ 0 & 0 & 1 \end{array}\right] \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & \hat{T}_{C_1} \\ r_{21} & r_{22} & r_{23} & \hat{T}_{C_2} \\ 0 & 0 & 0 & 1 \end{array}\right] \tilde{X} \\
&= \hat{K} \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & \hat{T}_{C_1} \\ r_{21} & r_{22} & r_{23} & \hat{T}_{C_2} \\ 0 & 0 & 0 & 1 \end{array}\right] \tilde{X}.
\end{aligned}$$

Notice that the calibration matrix $\hat{K}$ does not include the coordinates of the principal point u. The matrix $K_{1:2,1:2}$ is positive definite by construction.

□

Proposition 1.5 *An affine camera corresponds to an affine transformation A_{3D} in the 3D space, followed by an orthographic projection O_{ortho} to the image plane, followed by an affine transformation A_{2D} on the image plane.*

Proof: It follows from the following derivation,

$$\tilde{u} = (A_{2D} \cdot O_{ortho} \cdot A_{3D}) \cdot \tilde{X}$$

[1] We use a Matlab-like notation to indicate sub portions of matrices of vectors. For instance $A_{1:n,1:m}$ are the first n rows and m columns of A.

$$
\begin{aligned}
&= \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{X} \\
&= \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{X} \\
&= P_{aff} \tilde{X}.
\end{aligned}
$$

□

Notice than the A_{2D} transformation is basically related to the camera calibration matrix.

1.2 Planar affine camera

If we restrict the viewing conditions, we can additionally simplify the affine camera model. We suppose that the camera is viewing a planar scene, so that, without loss of generality, we can drop one of the three coordinates of the 3D space after having appropriately adjusted the matrix R to take into account the rotation of the viewing plane (we indicate with $\hat{R} = [\hat{r}_{ij}]$ the new rotation matrix). We consider the case in which we have $X_3 = 0$. Following the result of Proposition (16), we can plug the rotation $\hat{R}$, remove X_3, and we get the *planar affine camera* model,

$$
\tilde{u} = \hat{K} \left[\begin{array}{cc|c} \hat{r}_{11} & \hat{r}_{12} & \hat{T}_{C_1} \\ \hat{r}_{21} & \hat{r}_{22} & \hat{T}_{C_2} \\ 0 & 0 & 1 \end{array}\right] \begin{bmatrix} X_1 \\ X_2 \\ 1 \end{bmatrix}. \tag{16}
$$

that can be further rewritten as,

$$
\begin{aligned}
u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} &= \left[\begin{array}{cc} \alpha_1 & \alpha_s \\ 0 & \alpha_2 \end{array}\right] \left[\begin{array}{cc|c} \hat{r}_{11} & \hat{r}_{12} & \hat{T}_{C_1} \\ \hat{r}_{21} & \hat{r}_{22} & \hat{T}_{C_2} \end{array}\right] \begin{bmatrix} X_1 \\ X_2 \\ 1 \end{bmatrix} && (17) \\
&= \hat{K}_{1:2,1:2} [\hat{R}_{1:2,1:2} | \hat{T}_{C_{1:2}}] \begin{bmatrix} X_1 \\ X_2 \\ 1 \end{bmatrix} && (18) \\
&= \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ 1 \end{bmatrix} && (19)
\end{aligned}
$$

leading to a 2D to 2D model with 6 degrees of freedom.

How can we interpret the role of those degrees of freedom? If we forget about the camera model and we consider the effects of (19) on the image plane $\mathcal{X}$, in Figure 2 we report a visual interpretation of the 6 degrees of freedom. On the other hand, with the aim of understanding the transformations that were applied to the plane in the 3D space and that leaded to the corresponding pixels on $\mathcal{X}$, we can give another interpretation to (19). We rewrite (19) as

$$
u = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} p_{13} \\ p_{23} \end{bmatrix} = A \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + Z_C. \tag{20}
$$

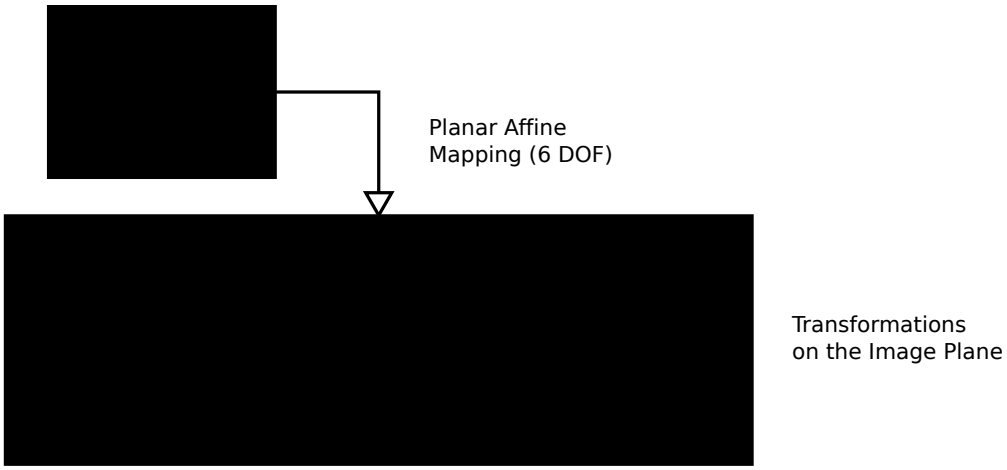

Figure 2: Planar affine imaging: transformation on the image plane $\mathcal{X}$ corresponding to the 6 degrees of freedom (translation is 2 degrees).

The vector $Z_C = [Z_{C_1}, Z_{C_2}]' = \hat{K}_{1:2,1:2}\hat{T}_{C_{1:2}}$ is a translation expressed in the camera frame $\perp_C$. It is related but *not* correspondent to the translation $\hat{T}_{C_{1:2}}$ of (17) from the optical center to the origin of $\perp_W$, due to the effect of the intrinsic camera parameters. The matrix $A = \hat{K}_{1:2,1:2}\hat{R}_{1:2,1:2}$ is a 2×2 positive define matrix, since $\hat{K}_{1:2,1:2}$ is positive definite by construction and $\hat{R}_{1:2,1:2}$ is the 2nd leading principal minor of a 3D rotation matrix, so that it is positive definite (since a rotation matrix is such that $\det(R) = 1$).

Now, using classic arguments from spectral theory and linear algebra, we can decompose the matrix A [3].

Theorem 1.1 *A positive 2×2 matrix A that defines an affine map which is not a similarity has a unique decomposition*

$$A = \lambda \cdot R_{\varphi_1} U_{\varphi_2} R_{\varphi_3} \tag{21}$$

where $\lambda > 0$ and R_{φ_1}, U_{φ_2}, R_{φ_3} are the following 2×2 matrices,

$$R_{\varphi_1} = \begin{bmatrix} \cos\varphi_1 & -\sin\varphi_1 \\ \sin\varphi_1 & \cos\varphi_1 \end{bmatrix} \tag{22}$$

$$U_{\varphi_2} = \begin{bmatrix} \frac{1}{\cos\varphi_2} & 0 \\ 0 & 1 \end{bmatrix} \tag{23}$$

$$R_{\varphi_3} = \begin{bmatrix} \cos\varphi_3 & -\sin\varphi_3 \\ \sin\varphi_3 & \cos\varphi_3 \end{bmatrix}, \tag{24}$$

with $\varphi_1 \in [0, 2\pi]$, $\varphi_2 \in [0, \frac{\pi}{2})$, and $\varphi_3 \in [0, \pi)$. In the case of a similarity transformation, we have $\varphi_2 = 0$.

Proof: The matrix $A'A$ is a symmetric positive semidefinite matrix, and it can be decomposed as $A'A = ODO'$, where O is an orthogonal transform (i.e. $O' = O^{-1}$) and D is a diagonal matrix of positive eigenvalues. We define $B = AOD^{-\frac{1}{2}}$, so that $BB' = AOD^{-\frac{1}{2}}D^{-\frac{1}{2}}O'A' = AOD^{-1}O'A' = A(A'A)^{-1}A' = I$, and we can conclude that B is an orthogonal transform. It is easy to check that

$$BD^{\frac{1}{2}}O' = AOD^{-\frac{1}{2}}D^{\frac{1}{2}}O' = AOO' = A$$

and, once we define $E = D^{\frac{1}{2}}$, we have

$$A = \lambda \cdot BEO'$$

where B and O are orthogonal matrices and E is a diagonal matrix with positive entries. Since $\det(A) > 0$, both $\det(B)$ and $\det(O)$ must share the same sign. Without any loss of generality we consider the subgroup of orthogonal transformations with unitary determinant (i.e. we rescale E). If $\det(B) = \det(O) = 1$, then B and O are rotations of angles φ_1 and $-\varphi_3$. If their determinant is -1, we can premultiply them by $\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$. We consider $\varphi_3 \in [0, \pi)$ (if $\varphi_3 \geq \pi$ we can replace φ_3 with $\varphi_3 - \pi$ and add π to φ_1). Moreover, $E = \begin{bmatrix} \lambda_0 & 0 \\ 0 & \lambda \end{bmatrix}$, with $\lambda_0 \geq \lambda$. We can group λ to get $E = \lambda \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix}$ with $t = \frac{\lambda_0}{\lambda} \geq 1$. Once we replace t with $\frac{1}{\cos\varphi_2}$, $\varphi_2 \in [0, \frac{\pi}{2})$ we get (21).

□

The angle φ_3 describes the rotation of the viewing plane around its normal (longitude). The angle φ_2 is between the normal to the viewing plane and the optical axis (latitude), whereas φ_1 is the camera spin around its optical axis. The parameter λ measures the zoom level (the camera can move forward and backward). See Figure 3. Notice that the matrix U_{φ_2} is a *tilt* transformation

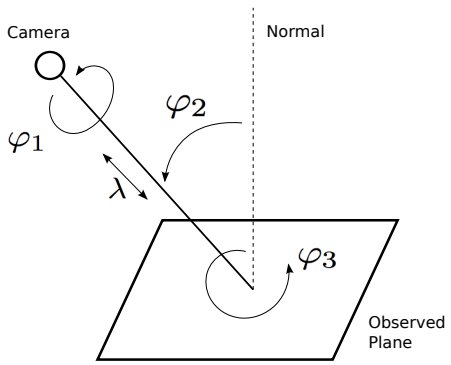

Figure 3: Affine camera viewing a plane, as described by (21).

of angle φ_2 along the first coordinate. It scales the first coordinate while keeping fixed the second one.

2 Digital image

The analysis of Section 1 implicitly considers continuos pixel coordinates $u = [u_1, u_2]'$ and an infinite resolution image plane $\mathcal{X}$. In order to move towards a discrete representation of the digital image, we have to introduce an additional layer of computations. Moreover, we must also take into account the optical blur of the camera lens. We indicate with z_d a digital image, and with z_c its continuos counterpart on $\mathcal{X}$, so that $z_c(u_1, u_2)$ is the pixel value at coordinates u. We indicate with $z_d(u_x, u_y)$ the pixel value at the discrete coordinates $(u_x, u_y) \in \mathbb{Z}^2$. The relationship between z_u and z_d can be modeled by

$$z_d = S_1 \mathcal{G}_{\sigma_1} z_c \tag{25}$$

where $\mathcal{G}_{\sigma_1}$ is a Gaussian convolution with standard deviation σ_1 (due to the optical blur of the camera lens), and S_1 is a sampling operator on a regular grid with mesh 1. The value of σ_1 is assumed to be selected such that there is no aliasing in the S_1 sampling process. This means that, if we are given the Shannon-Whittaker interpolation operator $\mathcal{S}$, defined as

$$(\mathcal{S}z_d)(u_1,u_2) = \sum_{(u_x,u_y)\in\mathbb{Z}^2} \mathrm{sinc}(u_1-u_x)\mathrm{sinc}(u_2-u_y), \tag{26}$$

where $\mathrm{sinc}x = \frac{\sin \pi x}{\pi x}$, we have

$$\mathcal{S}S_1\mathcal{G}_{\sigma_1}z_c = \mathcal{G}_{\sigma_1}z_c. \tag{27}$$

It is easy to check that $\mathcal{S}S_1 z_d = z_d$. From a practical point of view assuming that a given digital image has undergone a Gaussian blur $\sigma_1 \in [0.5, 1.0]$ is enough to prevent significant antialiasing.

Looking at (20) and (25) we can sketch the complete model that maps the transformation from a planar surface to the digital image by means of an affine camera. We define with p_c the "real-world" plane that we are viewing by the camera, and with $\mathcal{A}$ an affine map as in (20). It immediately follows that $z_c = \mathcal{A}p_c$, and, from (25), our complete mapping from the viewing plane to the digital image becomes

$$z_d = S_1\mathcal{G}_{\sigma_1}\mathcal{A}p_c. \tag{28}$$

2.1 Transforming the digital image

What if we change the parameters of the transformation $\mathcal{A}$? Does it correspond to applying the same transformation to z_d? The answer strictly depends on the type of transformation that we apply. In what follows we make use of continuous operators and in order to apply them to the discrete image z_d we must first use the $\mathcal{S}$ operator on z_d.

- **Translation**
  Suppose that $\mathcal{A}$ is simply a translation $\mathcal{T}$ (i.e. p_c is parallel to $\mathcal{X}$). It is easy to verify that $\mathcal{T}$ commute with $\mathcal{G}_{\sigma_1}$. If we indicate with τ_1 and τ_2 the translation offsets, we get

$$\begin{aligned}
(\mathcal{G}_{\sigma_1}\mathcal{T}p_c)(a_1,a_2) &= \int\int \frac{1}{2\pi\sigma_1^2} e^{-\frac{(a_1-\eta_1)^2+(a_2-\eta_2)^2}{2\sigma_1^2}} p_c(\eta_1+\tau_1,\eta_2+\tau_2)d\eta_1 d\eta_2 \\
&= \int\int \frac{1}{2\pi\sigma_1^2} e^{-\frac{(a_1+\tau_1-\gamma_1)^2+(a_2+\tau_2-\gamma_2)^2}{2\sigma_1^2}} p_c(\gamma_1,\gamma_2)d\gamma_1 d\gamma_2 \\
&= (\mathcal{G}_{\sigma_1}p_c)(a_1+\tau_1,a_2+\tau_2) \\
&= (\mathcal{T}\mathcal{G}_{\sigma_1}p_c)(a_1,a_2).
\end{aligned}$$

  We will use the commutativity of $\mathcal{T}$ and $\mathcal{G}_{\sigma_1}$ together with (27) in the following computations. In order to apply a translation $\mathcal{T}$ to z_d we must first rebuild the continuos image, then apply $\mathcal{T}$ and finally sample with S_1,

$$\begin{aligned}
\mathcal{T}z_d &:= S_1\mathcal{T}\mathcal{S}z_d \\
&= S_1\mathcal{T}\mathcal{S}S_1\mathcal{G}_{\sigma_1}p_c \\
&= S_1\mathcal{T}\mathcal{G}_{\sigma_1}p_c \\
&= S_1\mathcal{G}_{\sigma_1}\mathcal{T}p_c
\end{aligned}$$

  and we get a translation of the viewing plane (discarding border effects).

- **Rotation**
  Suppose that $\mathcal{A}$ is a rotation $\mathcal{R}$. Similarly to the previous case we can use the commutativity of $\mathcal{R}$ and $\mathcal{G}_{\sigma_1}$ together with (27) (the commutativity can be verified using the same procedure followed in the case of translations). We have

$$\begin{aligned} \mathcal{R}z_d &:= S_1\mathcal{R}\mathcal{S}z_d \\ &= S_1\mathcal{R}\mathcal{S}S_1\mathcal{G}_{\sigma_1}p_c \\ &= S_1\mathcal{R}\mathcal{G}_{\sigma_1}p_c \\ &= S_1\mathcal{G}_{\sigma_1}\mathcal{R}p_c \end{aligned}$$

  and we get a rotation of the viewing plane.

- **Zoom**
  Suppose that $\mathcal{A}$ is a uniform zoom $\mathcal{H}_\lambda$ of a factor λ ($\lambda > 1$ is a zoom out, $\lambda < 1$ corresponds to a zoom in). This case differs from the previous ones. Let us try to commute $\mathcal{G}_{\sigma_1}$ with $\mathcal{H}_\lambda$. We get

$$\begin{aligned} (\mathcal{G}_{\sigma_1}\mathcal{H}_\lambda p_c)(a_1,a_2) &= \int\int \frac{1}{2\pi\sigma_1^2} e^{-\frac{(a_1-\eta_1)^2+(a_2-\eta_2)^2}{2\sigma_1^2}} p_c(\lambda\eta_1,\lambda\eta_2)d\eta_1 d\eta_2 \\ &= \int\int \frac{1}{2\pi\sigma_1^2\lambda^2} e^{-\frac{(a_1-\gamma_1/\lambda)^2+(a_2-\gamma_2/\lambda)^2}{2\sigma_1^2}} p_c(\gamma_1,\gamma_2)d\gamma_1 d\gamma_2 \\ &= \int\int \frac{1}{2\pi\sigma_1^2\lambda^2} e^{-\frac{(\lambda a_1-\gamma_1)^2+(\lambda a_2-\gamma_2)^2}{2\sigma_1^2\lambda^2}} p_c(\gamma_1,\gamma_2)d\gamma_1 d\gamma_2 \\ &= (\mathcal{G}_{\sigma_1\lambda}p_c)(\lambda a_1,\lambda a_2) \\ &= (\mathcal{H}_\lambda\mathcal{G}_{\sigma_1\lambda}p_c)(a_1,a_2). \end{aligned}$$

  Commuting $\mathcal{G}_{\sigma_1}$ with $\mathcal{H}_\lambda$ causes an alteration of the standard deviation of the Gaussian kernel. Now, if we zoom the image z_d by a factor λ, we have

$$\begin{aligned} \mathcal{H}_\lambda z_d &:= S_1\mathcal{H}_\lambda\mathcal{S}z_d \\ &= S_1\mathcal{H}_\lambda\mathcal{S}S_1\mathcal{G}_{\sigma_1}p_c \\ &= S_1\mathcal{H}_\lambda\mathcal{G}_{\sigma_1}p_c \\ &= S_1\mathcal{G}_{\frac{\sigma_1}{\lambda}}\mathcal{H}_\lambda p_c \end{aligned} \tag{29}$$

  so that a zooming operation on z_d corresponds with the same operation applied to p_c with a different Gaussian blur. If we replace the original zoom with a zoom on z_d blurred by $\sigma_1\sqrt{\lambda^2-1}$ we get

$$\begin{aligned} \mathcal{H}_\lambda\mathcal{G}_{\sigma_1\sqrt{\lambda^2-1}}z_d &:= S_1\mathcal{H}_\lambda\mathcal{G}_{\sigma_1\sqrt{\lambda^2-1}}\mathcal{S}z_d \\ &= S_1\mathcal{H}_\lambda\mathcal{G}_{\sigma_1\sqrt{\lambda^2-1}}\mathcal{S}S_1\mathcal{G}_{\sigma_1}p_c \\ &= S_1\mathcal{H}_\lambda\mathcal{G}_{\sigma_1\sqrt{\lambda^2-1}}\mathcal{G}_{\sigma_1}p_c \\ &= S_1\mathcal{G}_{\sqrt{\sigma_1^2-\frac{\sigma_1^2}{\lambda^2}}}\mathcal{H}_\lambda\mathcal{G}_{\sigma_1}p_c \\ &= S_1\mathcal{G}_{\sqrt{\sigma_1^2-\frac{\sigma_1^2}{\lambda^2}}}\mathcal{G}_{\frac{\sigma_1}{\lambda}}\mathcal{H}_\lambda p_c \\ &= S_1\mathcal{G}_{\sigma_1}\mathcal{H}_\lambda p_c \end{aligned}$$

The last equation states that a zoom applied to the viewing plane p_c corresponds to a digital image z_d that is first blurred by $\sigma_1\sqrt{\lambda^2-1}$ and then zoomed by $\mathcal{H}_\lambda$. Setting $\lambda > 1$ realizes a zoom-out operation, i.e. some details are lost. The previous equation indicates that in order to recreate a zoom-out of the plane p_c and acquire a digital image using the same lens (σ_1), we have to take the digital image z_d, add Gaussian blur, and zoom-out the image. If $\lambda < 1$, we gets zoom-in operation, i.e. we see more details. In order to recreate the zoom-in effect, we should de-blur z_d, and zoom-in the image. This can be appreciated by the negative variance of the blur operator, $\sigma_1^2 - \frac{\sigma_1^2}{\lambda^2} < 0$.

- **Tilt**
  If $\mathcal{A}$ is a tilt $\mathcal{U}_\varphi$ along one axis we end up in a similar case to the previous one, since $\mathcal{U}_\varphi$ is a zoom along one coordinate only (suppose the first one). Again, using the same arguments of $\mathcal{H}_\lambda$, it can be shown that it does not commute with $\mathcal{G}_{\sigma_1}$. In particular, using Gaussian separability, it is easy to get the following relationship

$$\begin{aligned}
\mathcal{U}_\varphi z_d &:= S_1\mathcal{U}_\varphi\mathcal{S}z_d \\
&= S_1\mathcal{U}_\varphi\mathcal{S}S_1\mathcal{G}_{\sigma_1}p_c \\
&= S_1\mathcal{U}_\varphi\mathcal{G}_{\sigma_1}p_c \\
&= S_1(\mathcal{G}^{u_1}_{\frac{\sigma_1}{h}}\cdot\mathcal{G}^{u_2}_{\sigma_1})\otimes\mathcal{U}_\varphi p_c \\
&= S_1\mathcal{G}^{u_1}_{\frac{\sigma_1}{h}}(\mathcal{G}^{u_2}_{\sigma_1}\mathcal{U}_\varphi p_c)
\end{aligned}$$

  where $h = \frac{1}{\cos\varphi}$. A tilt in the u_x direction of z_d corresponds to the same operation applied to p_c (on u_1) followed by an anisotropic Gaussian blurring that blurs the tilted axis with a different standard deviation. If we replace the original tilt with a tilt on z_d blurred by $\sigma_1\sqrt{h^2-1}$ along u_x we get

$$\begin{aligned}
\mathcal{U}_\varphi\mathcal{G}^{u_x}_{\sigma_1\sqrt{h^2-1}}z_d &:= S_1\mathcal{U}_\varphi\mathcal{G}^{u_1}_{\sigma_1\sqrt{h^2-1}}\mathcal{S}z_d \\
&= S_1\mathcal{U}_\varphi\mathcal{G}^{u_1}_{\sigma_1\sqrt{h^2-1}}\mathcal{S}S_1\mathcal{G}_{\sigma_1}p_c \\
&= S_1\mathcal{U}_\varphi\mathcal{G}^{u_1}_{\sigma_1\sqrt{h^2-1}}\mathcal{G}_{\sigma_1}p_c \\
&= S_1\mathcal{G}^{u_1}_{\sqrt{\sigma_1^2-\frac{\sigma_1^2}{h^2}}}\mathcal{U}_\varphi\mathcal{G}_{\sigma_1}p_c \\
&= S_1\mathcal{G}^{u_1}_{\sqrt{\sigma_1^2-\frac{\sigma_1^2}{h^2}}}\mathcal{G}^{u_1}_{\frac{\sigma_1}{h}}(\mathcal{G}^{u_2}_{\sigma_1}\mathcal{U}_\varphi p_c) \\
&= S_1\mathcal{G}^{u_1}_{\sigma_1}(\mathcal{G}^{u_2}_{\sigma_1}\mathcal{U}_\varphi p_c) \\
&= S_1\mathcal{G}_{\sigma_1}\mathcal{U}_\varphi p_c
\end{aligned}$$

  The last equation states that a tilt $\mathcal{U}_\varphi$ applied to the viewing plane p_c corresponds to a digital image z_d that is first blurred along u_x by $\sigma_1\sqrt{\frac{1}{(\cos\varphi)^2}-1}$ and then tilted by $\mathcal{U}_\varphi$.

  A natural property of $\mathcal{U}_\varphi$ is that it can be converted into a tilt along the second coordinate $\underline{\mathcal{U}}_\varphi$ by means of a rotation,

$$\mathcal{R}_{(-\frac{\pi}{2})}\mathcal{U}_\varphi\mathcal{R}_{\frac{\pi}{2}} = \underline{\mathcal{U}}_\varphi. \tag{30}$$

Finally, we notice that the previously described operators commute (at least in a weak sense), so that the reported analysis can be easily extended to the case in which a rotation, a translation, a zoom, and a tilt are combined.

2.1.1 Scale space

Building the scale space of an image z_d consists in computing the function $s(\sigma, z_d)$, which returns a digital image corresponding to z_d blurred by a Gaussian kernel of width σ, for increasing values of σ. More formally, if we define $\sigma(i)$ as the function that returns the i-th scale,

$$s(\sigma(i), z_d) = \mathcal{G}_{\sigma(i)} z_d, \quad i = 0, \ldots, q-1. \tag{31}$$

Given $\sigma(0) = \sigma$, we write $\sigma(i) = k_i \sigma(0)$, and, following (29), the scale space can be equivalently defined in terms of variable zoom operations of a factor k_i and using a fixed kernel $\mathcal{G}_\sigma$,

$$\begin{aligned} s(k_i\sigma(0), z_d) &= \mathcal{G}_{k_i\sigma(0)} z_d \\ &= \mathcal{H}_{\frac{1}{k_i}} \mathcal{H}_{k_i} \mathcal{G}_{k_i\sigma(0)} z_d \\ &= \mathcal{H}_{\frac{1}{k_i}} \mathcal{G}_{\sigma(0)} \mathcal{H}_{k_i} z_d \end{aligned}$$

so that the scale space can be interpreted as a zoom out of z_d by k_i, blur by a fixed kernel of width $\sigma(0)$, zoom in of the result by k_i.

We can extend this definition including the mean $m \neq 0$ of $\mathcal{G}_\sigma$, that becomes $\mathcal{G}^m_\sigma$,

$$(\mathcal{G}^m_\sigma z_d)(u_x, u_y) := \frac{1}{2\pi\sigma^2} \int\int e^{-\frac{(m_x+u_x-\tau_x)^2+(m_y+u_y-\tau_y)^2}{2\sigma^2}} z_d(\tau_x, \tau_y) d\tau_x d\tau_y$$

If we apply a zoom operation $\mathcal{H}_\lambda$, we have

$$\begin{aligned} (\mathcal{G}^m_\sigma \mathcal{H}_\lambda z_d)(u_x, u_y) &:= \frac{1}{2\pi\sigma^2} \int\int e^{-\frac{(m_x+u_x-\tau_x)^2+(m_y+u_y-\tau_y)^2}{2\sigma^2}} z_d(\lambda\tau_x, \lambda\tau_y) d\tau_x d\tau_y \\ &= \frac{1}{2\pi(\lambda\sigma)^2} \int\int e^{-\frac{(\lambda m_x+\lambda u_x-\gamma_x)^2+(\lambda m_y+\lambda u_y-\gamma_y)^2}{2(\lambda\sigma)^2}} z_d(\gamma_x, \gamma_y) d\gamma_x d\gamma_y \\ &= \mathcal{H}_\lambda \mathcal{G}^{\lambda m}_{\lambda\sigma}(u_x, u_y), \end{aligned}$$

that is the analogous of equation (29) when considering the mean of the Gaussian,

$$\mathcal{G}^m_\sigma \mathcal{H}_\lambda z_d = \mathcal{H}_\lambda \mathcal{G}^{\lambda m}_{\lambda\sigma} z_d. \tag{32}$$

Given $\sigma(0) = \sigma$, $m(0) = m$, the extended scale space definition becomes

$$\begin{aligned} s(k_i\sigma(0), k_i m(0), z_d) &= \mathcal{G}^{k_i m(0)}_{k_i\sigma(0)} z_d \\ &= \mathcal{H}_{\frac{1}{k_i}} \mathcal{H}_{k_i} \mathcal{G}^{k_i m(0)}_{k_i\sigma(0)} z_d \\ &= \mathcal{H}_{\frac{1}{k_i}} \mathcal{G}^{m(0)}_{\sigma(0)} \mathcal{H}_{k_i} z_d, \end{aligned}$$

that is, filtering z_d with a Gaussian kernel of with $k_i\sigma$ and mean $k_i m$ corresponds with a zoom out of z_d by k_i, blur by a fixed kernel of width σ and centered in m, zoom in of the result by k_i.

Now, we recall that z_d is the discrete counterpart of the continuos p_c (forgetting about the planar transformations), that has been already blurred by σ_1. We redefine the scale space in order

to consider this aspect,

$$\begin{aligned} s(\sigma(i), z_d) &= S_1 \mathcal{G}_{\sigma(i)} p_c \\ &= S_1 \mathcal{G}_{\sqrt{\sigma(i)^2 - \sigma_1^2}} \mathcal{G}_{\sigma_1} p_c \\ &= S_1 \mathcal{G}_{\sqrt{\sigma(i)^2 - \sigma_1^2}} z_d \\ &= S_1 \mathcal{G}_{\sigma_r(i)} z_d \end{aligned}$$

where $\sigma_r(i) = \sqrt{\sigma(i)^2 - \sigma_1^2}$ is the *real* blur that we have to apply to z_d to a get the element of the scale space indexed by $\sigma(i)$.

If we assume $\sigma_1 = 0.5$, a practical results from related literature [2] suggest to build the scale space by first doubling the image size (i.e. $\mathcal{H}_{\frac{1}{2}} z_d$, linear interpolation), leading to an image with $\sigma_1 = 2 \times 0.5 = 1.0$ due to (29). Several experimentations show that $\sigma(0) = 1.6$ is a reasonable choice for feature extraction, so that, starting from the doubled image, $\sigma_r(i) \approx 1.24$. Other authors [3] suggests to use $\sigma_1 = 0.8$, so that, after doubling the image size, we get an image with $\sigma_1 = 2 \times 0.8 = 1.6$. In this case we have to set $\sigma_r(0) = 0$ to get $\sigma(0) = 1.6$, i.e. $s(\sigma(0), z_d)$ is simply the doubled image without any additional blurs.

We consider again the scale space in which $\sigma(i) = k_i \sigma(0)$. From (29) we have

$$\begin{aligned} s(k_i \sigma(0), z_d) &= S_1 \mathcal{H}_{\frac{1}{k_i}} \mathcal{G}_{\sigma(0)} \mathcal{H}_{k_i} p_c \\ &= \mathcal{H}_{\frac{1}{k_i}} (\mathcal{H}_{k_i} \mathcal{G}_{\sigma(0)\sqrt{k_i^2 - 1}} z_d) \\ &= \mathcal{H}_{\frac{1}{k_i}} \mathcal{G}_{\sigma(0)\sqrt{1 - \frac{1}{k_i^2}}} \mathcal{H}_{k_i} z_d \\ &= \mathcal{G}_{\sigma(0)\sqrt{k_i^2 - 1}} z_d \\ &= \mathcal{G}_{\sigma_r}(i) z_d \end{aligned} \tag{33}$$

In this setting the *real* blur to be applied to z_d is $\sigma_r(i) = \sigma(0)\sqrt{k_i^2 - 1}$.

Another commonly used property of scale spaces is that if we downscale the image $s(2\sigma(i), z_d)$ by a factor $\frac{1}{2}$, we get

$$\begin{aligned} \mathcal{H}_2 s(2\sigma(i), z_d) &= \mathcal{H}_2 \mathcal{G}_{2\sigma(i)} p_c \\ &= \mathcal{G}_{\sigma(i)} \mathcal{H}_2 p_c \\ &= \mathcal{G}_{\sigma_r(i)} \mathcal{G}_{\sigma_1} \mathcal{H}_2 p_c \end{aligned} \tag{34}$$

so that we get a zoomed-out p_c blurred by $\sigma(i)$.

2.2 Artificially generated transformations

Suppose that we are only given a digital image z_d and we want to artificially simulate the image $\hat{z}_d$ af the *same resolution* of z_d that corresponds to an affine transformation of the observed plane on the 3D space. From (20), Theorem 1.1, and the model of (25) we can write

$$\begin{aligned} \hat{z}_d = g(p_c) &:= S_1 \mathcal{H}_{\frac{1}{\lambda}} (\mathcal{G}_{\sigma_1} \mathcal{A} p_c) \\ &= S_1 \mathcal{H}_{\frac{1}{\lambda}} (\mathcal{G}_{\sigma_1} \mathcal{T} \mathcal{H}_\lambda \mathcal{R}_{\varphi_1} \mathcal{U}_{\varphi_2} \mathcal{R}_{\varphi_3} p_c). \end{aligned}$$

Using the previously described commutation properties we write $\hat{z}_d$ in function of z_d,

$$\hat{z}_d = g(z_d) \quad = \quad \mathcal{H}_{\frac{1}{\lambda}}(\mathcal{T}\mathcal{H}_\lambda \mathcal{G}_{\sigma_1\sqrt{\lambda^2-1}} \mathcal{R}_{\varphi_1} \mathcal{U}_{\varphi_2} \mathcal{G}^{u_x}_{\sigma_1\sqrt{\frac{1}{(cos\varphi_2)^2}-1}} \mathcal{R}_{\varphi_3} z_d).$$

Translation $\mathcal{T}$ and zoom $\mathcal{H}$ weakly commute, in the sense that there exists a translation $\underline{\mathcal{T}}$ for which $\underline{\mathcal{T}}\mathcal{H} = \mathcal{H}\mathcal{T}$. We get

$$\hat{z}_d = g(z_d) \quad = \quad \underline{\mathcal{T}}\mathcal{H}_{\frac{1}{\lambda}} \mathcal{H}_\lambda \mathcal{G}_{\sigma_1\sqrt{\lambda^2-1}} \mathcal{R}_{\varphi_1} \mathcal{U}_{\varphi_2} \mathcal{G}^{u_x}_{\sigma_1\sqrt{\frac{1}{(cos\varphi_2)^2}-1}} \mathcal{R}_{\varphi_3} z_d.$$

Now, suppose that the optical axis is aligned with the observed point (i.e. no translation), then

$$\begin{aligned}
\hat{z}_d = g(z_d) &= \mathcal{H}_{\frac{1}{\lambda}} \mathcal{H}_\lambda \mathcal{G}_{\sigma_1\sqrt{\lambda^2-1}} \mathcal{R}_{\varphi_1} \mathcal{U}_{\varphi_2} \mathcal{G}^{u_x}_{\sigma_1\sqrt{\frac{1}{(cos\varphi_2)^2}-1}} \mathcal{R}_{\varphi_3} z_d \\
&= \mathcal{H}_{\frac{1}{\lambda}} \mathcal{G}_{\sigma_1\sqrt{1-\frac{1}{\lambda^2}}} \mathcal{H}_\lambda \mathcal{R}_{\varphi_1} \mathcal{U}_{\varphi_2} \mathcal{G}^{u_x}_{\sigma_1\sqrt{\frac{1}{(cos\varphi_2)^2}-1}} \mathcal{R}_{\varphi_3} z_d \\
&= \mathcal{R}_{\varphi_1} \mathcal{H}_{\frac{1}{\lambda}} \mathcal{G}_{\sigma_1\sqrt{1-\frac{1}{\lambda^2}}} \mathcal{H}_\lambda \mathcal{U}_{\varphi_2} \mathcal{G}^{u_x}_{\sigma_1\sqrt{\frac{1}{(cos\varphi_2)^2}-1}} \mathcal{R}_{\varphi_3} z_d
\end{aligned}$$

Comparing the last expression with (33), it is easy to see that we can write the above formula as

$$\hat{z}_d = \mathcal{R}_{\varphi_1} s \left(\lambda\sigma_1, \quad \mathcal{U}_{\varphi_2} \mathcal{G}^{u_x}_{\sigma_1\sqrt{\frac{1}{(cos\varphi_2)^2}-1}} \mathcal{R}_{\varphi_3} z_d \right).$$

that leads to the following proposition.

Proposition 2.1 *We are given a plane in a 3D space, an affine camera with blur σ_1, and the digital image z_d acquired by the camera. A transformation of the plane, parametrized by the Euler's angles $\varphi_1, \varphi_2, \varphi_3$ and by the zoom factor λ, generates a digital image $\hat{z}_d$ that can be equivalently obtained by appropriately transforming z_d. In particular, $\hat{z}_d$ is the element indexed by $\lambda\sigma_1$ in the scale space of $\mathcal{U}_{\varphi_2} \mathcal{G}^{u_x}_{\sigma_1\sqrt{\frac{1}{(cos\varphi_2)^2}-1}} \mathcal{R}_{\varphi_3} z_d$ rotated by φ_1,*

$$\hat{z}_d = \mathcal{R}_{\varphi_1} s \left(\sigma, \quad \mathcal{U}_{\varphi_2} \mathcal{G}^{u_x}_{\sigma_1\sqrt{\frac{1}{(cos\varphi_2)^2}-1}} \mathcal{R}_{\varphi_3} z_d \right), \tag{35}$$

where $\sigma = \lambda\sigma_1$ and $\mathcal{R}_{\varphi_1}$, $\mathcal{R}_{\varphi_3}$, $\mathcal{U}_{\varphi_2}$ are the continuos operators corresponding to R_{φ_1}, R_{φ_3}, U_{φ_2} from Theorem 1.1.

2.2.1 Sampling the parameters

Suppose that we are given z_d and want to define the discrete sets of parameters $\varphi_1, \varphi_2, \varphi_3, \sigma$ that can be used to reasonably approximate the projections corresponding to all the possible transformations in the 3D space of the observed object (plane).

From (35) the first transformation that we have to apply to z_d is $\mathcal{R}_{\varphi_3}$ followed by $\mathcal{U}_{\varphi_2} \mathcal{G}^{u_x}_{\sigma_1\sqrt{\frac{1}{(cos\varphi_2)^2}-1}}$.

From Theorem 1.1 we have $\varphi_2 \in [0, \frac{\pi}{2})$ and $\varphi_3 \in [0, \pi)$. From a practical point of view it is enough to set $\varphi_2 \in [0, 80°]$. The image distortion caused by a small rotation $\mathcal{R}_{\varphi_3}$ is more evident for larger tilts $\mathcal{U}_{\varphi_2}$. We have to sample φ_3 at a finer grain when φ_2 increases. Experimental evidence

Table 1: Affine transformation. Parameters and discrete sampling.

Parameter	Range	Sampling Precision	Type
σ	$[\sigma_1, \infty)$	$\Delta\sigma = 2^{\frac{1}{p}}$, $p = 3$	RATIO
φ_1	$[0, 2\pi)$	$\Delta\varphi_1 = 15^\circ$	OFFSET
φ_2	$\left[0, \frac{\pi}{2}\right)$	$\Delta h = \Delta\left(\frac{1}{\cos\varphi 2}\right) = \sqrt{2}$	RATIO
φ_3	$[0, 80^\circ]$	$\Delta\varphi_3 = \frac{72^\circ}{h} = 72^\circ \cos\varphi_2$	OFFSET

suggests that an angle displacement of $\Delta\varphi_3 = \frac{72^\circ}{h} = 72^\circ \cos\varphi_2$ is enough [3]. The angle φ_2 must be sampled considering that when it is large, a small variation will generate a more evident distortion on the image plane. For this reason, following [3], we select the ratio Δh between two consecutive values of $h = \frac{1}{\cos\varphi_2}$ as $\Delta h = \sqrt{2}$. The maximum value of h is approximately $4\sqrt{2}$. The following transformation is $\mathcal{R}_{\varphi_1}$ that is an in-plane rotation. It can be uniformly sampled with an offset $\Delta\varphi_1 = 15^\circ$. The final transformation is a scale change. Given an image z_d, with prior blurring σ_1, we have to define an initial $\sigma(0)$ and build the scale space of element equally spaced by the factor $\Delta\sigma$, For efficiency, it is reasonable to halve the image size for every doubling of $\sigma(0)$. We divide the computation into several octaves (i.e. doubling of $\sigma(0)$) of scale space elements separated by a factor $\Delta\sigma = 2^{\frac{1}{p}}$, where p is the number of intervals in each octave.

In detail, we start by computing $s(\sigma(0), z_d)$ with $\sigma(0) = 1.6$, and $\sigma_r(0) = \sqrt{\sigma(0) - \sigma_1}$, as suggested in [2]. Then we can build the first octave by $s(\sigma(i) = k_i\sigma(0), z_d)$, $k_i = 2^{\frac{i}{p}}$, $i = 1, \ldots, p$, so that two consecutive elements are spaced by $\Delta\sigma$ in the scale space. Then, we halve the last element of the octave, that is $s(2\sigma(0), z_d)$, by skipping every second pixel. This allow us to get a zoomed out view z_d^{small} of the viewing plane blurred by $\sigma(0)$ (34). Now we set $z_d \leftarrow z_d^{small}$, and we build the next octave $s(\sigma(i) = k_i\sigma(0), z_d)$, $k_i = 2^{\frac{i}{p}}$, $i = 1, \ldots, p$ and so on. Experimental evidence [2] shows that sampling $p = 3$ scales on each octave is enough for the purpose of feature extraction from a pixel neighborhood. Notice that in order to generate $s(\sigma(i), z_d)$ it is sufficient to blur $s(\sigma(i-1), z_d)$ by $\mathcal{G}_{\sigma(i-1)\sqrt{\Delta\sigma^2 - 1}}$.

3 Affine invariant filter functions

Considering the camera model of Section 1 and the digital image model of Section 2, we study a family of filter functions that operates on the neighborhood of a pixel $u = [u_x, u_y] \in \mathbb{Z}^2$ belonging to a digital image z_d, $f : \mathbb{Z}^2 \to I\!R$. We can use the selected camera model once we consider that f operates on small regions of the input image and that we can approximate the observed object by the tangent plane to the object that passes on u (i.e. smooth object surfaces). We define a filter function as

$$f(u_x, u_y, \sigma, \varphi_1) = \frac{1}{2\pi\sigma^2\mu^2} \int\int \sum_{k=1}^{l} \alpha_k e^{-\frac{(\sigma(R_{-\varphi_1} m_k)_x + u_x - \tau_1)^2 + (\sigma(R_{-\varphi_1} m_k)_y + u_y - \tau_2)^2}{2\mu^2\sigma^2}} z_d(\tau_1, \tau_2) d\tau_1 d\tau_2 \tag{36}$$

where $\alpha_k \in I\!R$, the number of components l, the means m_k, and the value of μ are given in advance. Using the notation of Section 2, we can rewrite f as

$$\begin{aligned}
f(u_x, u_y, \sigma, \varphi_1) &= \sum_{k=1}^{l} \left(\mathcal{G}_{\sigma\mu}^{\sigma R_{-\varphi_1} m_k} z_d \right)(u_x, u_y) \\
&= \sum_{k=1}^{l} \left(\mathcal{H}_{\frac{1}{\sigma}} \mathcal{G}_{\mu}^{R_{-\varphi_1} m_k} \mathcal{H}_{\sigma} z_d \right)(u_x, u_y) \\
&= \sum_{k=1}^{l} \alpha_k \cdot s(\sigma\mu, \sigma R_{-\varphi_1} m_k, z_d)(u_x, u_y) \\
&= \sum_{k=1}^{l} \alpha_k \cdot \left[\mathcal{T}_{\sigma R_{-\varphi_1} m_k} s(\sigma, z_d) \right](u_x, u_y) \\
&= \sum_{k=1}^{l} \alpha_k \cdot \left[\mathcal{T}_{\sigma m_k} \mathcal{R}_{\varphi_1} s(\sigma, z_d) \right](u_x, u_y) \\
&= \sum_{k=1}^{l} \alpha_k \cdot \xi_k(u_x, u_y, \sigma, \varphi_1) \\
&= < \alpha, \xi(u_x, u_y, \sigma, \varphi_1) > .
\end{aligned}$$

We assume that filters are designed at $\sigma = 1$ and that $\sigma \geq 1$. In other words, μ is the initial blur that we apply to the image z_d before starting the filtering process, and that filtering proceeds at scale multiple of μ. In the previous section we defined $\sigma(0)$ as the starting value for scale space construction, so that we have $\sigma(0) = \sqrt{\mu^2 + \sigma_1^2}$, since the continuos image is blurred by a prior blur of σ_1. As a consequence, $\mu = \sigma_r(0) = \sqrt{\sigma(0)^2 - \sigma_1^2}$. In the previous section we roughly estimated σ_1 to be in $[0.5, 1]$, and that $\sigma(0) = 1.6$ was a good starting value, so that $\mu \in [1.2, 1.5]$.

Now, from Proposition 2.1, we can extend the definition of f to handle projections of a 3D plane that incurs in affine transformations. We get

$$\begin{aligned}
f(u_x, u_y, \sigma, \varphi_1, \varphi_2, \varphi_3) &= \sum_{k=1}^{l} \alpha_k \cdot \left[\mathcal{T}_{\sigma m_k} \mathcal{R}_{\varphi_1} s \left(\sigma, \quad \mathcal{U}_{\varphi_2} \mathcal{G}_{\sigma_1 \sqrt{\frac{1}{(cos\varphi_2)^2} - 1}}^{u_x} \mathcal{R}_{\varphi_3} z_d \right) \right](u_x, u_y) \\
&= \sum_{k=1}^{l} \alpha_k \cdot \left[\mathcal{T}_{\sigma m_k} \mathcal{R}_{\varphi_1} s \left(\sigma, z_d^{\varphi_2, \varphi_3} \right) \right](u_x, u_y) \\
&= \sum_{k=1}^{l} \alpha_k \cdot \xi_k(u_x, u_y, \sigma, \varphi_1, \varphi_2, \varphi_3) \\
&= < \alpha, \xi(u_x, u_y, \sigma, \varphi_1, \varphi_2, \varphi_3) > .
\end{aligned}$$

Remark 3.1 *Once we define a unique mechanism to select the values of σ and of the angles with respect to the observed scene, the filter function becomes $f(u_x, u_y)$ and it is an affine invariant filter that can handle invariances with respect to affine transformations of the projected object in the 3D space.*

3.1 Filters and different cameras

The function f does depend on the camera calibration matrix $\hat{K}$. This means that selecting, say, a given value of φ_1, does not correspond to a real rotation of φ_1 in the 3D space, since the rotation is influenced by $\hat{K}$. As a result, fixing the value of φ_1 (or one of the other parameters) will result in a different rotation of the 3D object on different cameras.

However, if we do not fix φ_1 in advance and we define a unique mechanism to select the value of φ_1 with respect to the observed scene, switching the camera only requires to recompute φ_1. We do not need to calibrate the camera to use the filter. On the other hand, if we want to recover the real rotation of the object, we need to estimate $\hat{K}$. The same considerations hold for the other parameters.

Non linear distortion $d(\cdot)$ due to the camera lens can be discarded, since we are assuming to operate on small regions of the image.

3.2 3D object location

Again, once we define a unique mechanism to select the values of σ and of the angles with respect to the observed scene, we can recover the rotation (Euler's angles) of the observed object and its scale. We do not have a clear information on its coordinates in the 3D space, since we should take into account the depth and the perspective correction.

The scale σ gives us a rough estimate of the object depth/size, that we can use to make inference on the depth map of the observed scene, once we consider the relative scales of the pixels belonging to the current image.

Techniques that are based on *structure from motion* (SFM) could be used to estimate the 3D coordinates on an input video, using the filter function f to localize corresponding pixels among different views of the same scene.

References

[1] M Gori, Gnecco G., Melacci S., and Sanguineti M. Learning from constraints. Technical report, Technical Report, University of Siena, 2012.

[2] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[3] J.M. Morel and G. Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.