3D geometry basics (for robotics) lecture notes

Marc Toussaint
Machine Learning & Robotics lab, FU Berlin
Arnimallee 7, 14195 Berlin, Germany

October 30, 2011

This document introduces to some basic geometry, focussing on 3D transformations, and introduces proper conventions for notation. There exist one-to-one implementations of the concepts and equations in libORS.

1 Rotations

There are many ways to represent rotations in $SO(3)$. We restrict ourselves to three basic ones: rotation matrix, rotation vector, and quaternion. The rotation vector is also the most natural representation for a "rotation velocity" (angular velocities). Euler angles or raw-pitch-roll are an alternative, but they have singularities and I don't recommend using them in practice.

A rotation matrix is a matrix $R \in \mathbb{R}^{3\times3}$ which is orthonormal (columns and rows are orthogonal unit vectors, implying determinant 1). While a 3×3 matrix has 9 degrees of freedom (DoFs), the constraint of orthogonality and determinant 1 constraints this: The set of rotation matrices has only 3 DoFs ($\sim$ the local Lie algebra is 3-dim).

The application of R on a vector x is simply the matrix-vector product Rx.

Concatenation of two rotations R_1 and R_2 is the normal matrix-matrix product R_1R_2.

Inversion is the transpose, $R^{-1} = R^\top$.

A rotation vector is an unconstraint vector $w \in \mathbb{R}^3$. The vector's direction $\underline{w} = \frac{w}{|w|}$ determines the rotation axis, the vector's length $|w| = \theta$ determins the rotation angle (in radians, using the right thumb convention).

The application of a rotation described by $w \in \mathbb{R}^3$ on a vector $x \in \mathbb{R}^3$ is given as (Rodrigues' formula)

$$w \cdot x = \cos\theta\, x + \sin\theta\, (\underline{w} \times x) + (1 - \cos\theta)\, \underline{w}(\underline{w}^\top x) \tag{1}$$

where $\theta = |w|$ is the rotation angle and $\underline{w} = w/\theta$ the unit length rotation axis.

The inverse rotation is described by the negative of the rotation vector.

Concatenation is non-trivial in this representation and we don't discuss it here. In practice, a rotation vector is first converted to a rotation matrix or quaternion.

Convertion to a matrix: For every vector $w \in \mathbb{R}^3$ we define its skew symmetric matrix as

$$\widehat{w} = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix} . \tag{2}$$

Note that such skew-symmetric matrices are related to the cross product: $w \times v = \widehat{w}\, v$, where the cross product is rewritten as a matrix product. The rotation matrix $R(w)$ that corresponds to a given rotation vector w is:

$$R(w) = \exp(\widehat{w}) \tag{3}$$

$$= \cos\theta\, I + \sin\theta\, \widehat{w}/\theta + (1 - \cos\theta)\, ww^\top/\theta^2 \tag{4}$$

The $\exp$ function is called exponential map (generating a group element (=rotation matrix) via an element of the Lie algebra (=skew matrix)). The other formular is called Rodrigues' formular: the first term is a diagonal matrix (I is the 3D identity matrix), the second terms the skew symmetric part, the last term the symmetric part ($ww^\top$ is also called outper product).

Angular velocity & derivative of a rotation matrix: We represent angular velocities by a vector $w \in \mathbb{R}^3$, the direction $\underline{w}$ determines the rotation axis, the length $|w|$ is the rotation velocity (in radians per second). When a body's orientation at time t is described by a rotation matrix $R(t)$ and the body's angular velocity is w, then

$$\dot{R}(t) = \widehat{w}\, R(t) . \tag{5}$$

(That's intuitive to see for a rotation about the x-axis with velocity 1.) Some insights from this relation: Since $R(t)$ must always be a rotation matrix (fulfill orthogonality and determinant 1), its derivative $\dot{R}(t)$ must also fulfill certain constraints; in particular it

can only live in a 3-dimensional sub-space. It turns out that the derivative $\dot{R}$ of a rotation matrix R must always be a skew symmetric matrix $\widehat{w}$ times R – anything else would be inconsistent with the contraints of orthogonality and determinant 1.

Note also that, assuming $R(0) = I$, the solution to the differential equation $\dot{R}(t) = \widehat{w}\, R(t)$ can be written as $R(t) = \exp(t\widehat{w})$, where here the exponential function notation is used to denote a more general so-called exponential map, as used in the context of Lie groups. It also follows that $R(w)$ from (3) is the rotation matrix you get when you rotate for 1 second with angular velocity described by w.

Quaternion (I'm not describing the general definition, only the "quaternion to represent rotation" definition.) A quaternion is a unit length 4D vector $r \in \mathbb{R}^4$; the first entry r_0 is related to the rotation angle θ via $r_0 = \cos(\theta/2)$, the last three entries $\bar{r} \equiv r_{1:3}$ are related to the unit length rotation axis $\underline{w}$ via $\bar{r} = \sin(\theta/2)\, \underline{w}$.

The inverse of a quaternion is given by negating $\bar{r}$, $r^{-1} = (r_0, -\bar{r})$ (or, alternatively, negating r_0).

The concatenation of two rotations r, r' is given as the quaternion product

$$r \circ r' = (r_0 r'_0 - \bar{r}^\top \bar{r}',\ r_0 \bar{r}' + r'_0 \bar{r} + \bar{r}' \times \bar{r}) \tag{6}$$

The application of a rotation quaternion r on a vector x can be expressed by converting the vector first to the quaternion $(0, x)$, then computing

$$r \cdot x = (r \circ (0, x) \circ r^{-1})_{1:3}\ , \tag{7}$$

I think a bit more efficient is to first convert the rotation quaternion r to the equivalent rotation matrix R, as given by

$$R = \begin{pmatrix} 1 - r_{22} - r_{33} & r_{12} - r_{03} & r_{13} + r_{02} \\ r_{12} + r_{03} & 1 - r_{11} - r_{33} & r_{23} - r_{01} \\ r_{13} - r_{02} & r_{23} + r_{01} & 1 - r_{11} - r_{22} \end{pmatrix}$$
$$r_{ij} := 2 r_i r_j\ . \tag{8}$$

(Note: In comparison to (3) this does not require to compute a $\sin$ or $\cos$.) Inversely, the quaterion r for a given matrix R is

$$r_0 = \frac{1}{2}\sqrt{1 + \mathrm{tr} R} \tag{9}$$
$$r_3 = (R_{21} - R_{12})/(4 r_0) \tag{10}$$
$$r_2 = (R_{13} - R_{31})/(4 r_0) \tag{11}$$
$$r_1 = (R_{32} - R_{23})/(4 r_0)\ . \tag{12}$$

Angular velocity → quaternion velocity Given an angular velocity $w \in \mathbb{R}^3$ and a current quaterion $r(t) \in \mathbb{R}$, what is the time derivative $\dot{r}(t)$ (in analogy to Eq. (5))? For simplicity, let's first assume $|w| = 1$. For a small time interval δ, w generates a rotation vector δw, which converts to a quaterion

$$\Delta r = (\cos(\delta/2), \sin(\delta/2) w)\ . \tag{13}$$

That rotation is concatenated LHS to the original quaternion,

$$r(t + \delta) = \Delta r \circ r(t)\ . \tag{14}$$

Now, if we take the derivative w.r.t. δ and evaluate it at $\delta = 0$, all the $\cos(\delta/2)$ terms become $-\sin(\delta/2)$ and evaluate to zero, all the $\sin(\delta/2)$ terms become $\cos(\delta/2)$ and evaluate to one, and we have

$$\dot{r}(t) = \frac{1}{2}(-w^\top \bar{r},\ r_0 w + \bar{r} \times w) = \frac{1}{2}(0, w) \circ r(t) \tag{15}$$

Here $(0, w) \in \mathbb{R}^4$ is a four-vector; for $|w| = 1$ it is a normalized quaternion. However, due to the linearity the equation holds for any w.

Quaternion velocity → angular velocity The following is relevant when taking the derivative w.r.t. the parameters of a quaternion, e.g., for a ball joint represented as quaternion. Given $\dot{r}$, we have

$$\dot{r} \circ r^{-1} = \frac{1}{2}(0, w) \circ r \circ r^{-1} = \frac{1}{2}(0, w) \tag{16}$$

which allows us to read off the angular velocity induced by a change of quaternion. However, the RHS zero will hold true only iff $\dot{r}$ is orthogonal to r (where $\dot{r}^\top r = \dot{r}_0 r_0 + \dot{\bar{r}}^\top \bar{r} = 0$, see (6)). In case $\dot{r}^\top r \neq 0$, the change in length of the quaterion does not represent any angular velocity; in typical kinematics engines a non-unit length is ignored. Therefore one first orthogonalizes $\dot{r} \leftarrow \dot{r} - r(\dot{r}^\top r)$.

As a special case of application, consider computing the partial derivative w.r.t. quaternion coordinates, where $\dot{r}$ is the unit vectors $e_0, .., e_3$. In this case, the orthogonalization becomes simply $e_i \leftarrow e_i - r r_i$ and

$$(e_i - r_i r) \circ r^{-1} = e_i \circ r^{-1} - r_i (1, 0, 0, 0) \tag{17}$$
$$w_i = 2 [e_i \circ r^{-1}]_{1:3}\ , \tag{18}$$

where w_i is the rotation vector implied by $\dot{r} = e_i$. In case the original quaternion r wasn't normalized (which could be, if a standard optimization algorithm searches in the quaternion configuration space), then r actually represents the normalized quaternion $\bar{r} = \frac{1}{\sqrt{r^2}} r$, and (due to linearity of the above), the rotation vector implied by $\dot{r} = e_i$ is

$$w_i = \frac{2}{\sqrt{r^2}}\, [e_i \circ r^{-1}]_{1:3}\ . \tag{19}$$

2 Transformations

We consider two types of transformations here: either static (translation+rotation), or dynamic (translation+velocity+rotation+angular velocity). The first maps

between two static reference frames, the latter between moving reference frames, e.g. between reference frames attached to moving rigid bodies.

2.1 Static transformations

Concerning the static transformations, again there are different representations:

A homogeneous matrix is a 4×4-matrix of the form

$$T = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \tag{20}$$

where R is a 3×3-matrix (rotation in our case) and t a 3-vector (translation).

In homogeneous coordinates, vectors $x \in \mathbb{R}^3$ are expanded to 4D vectors $\begin{pmatrix} x \\ 1 \end{pmatrix} \in \mathbb{R}^4$ by appending a 1.

Application of a transform T on a vector $x \in \mathbb{R}^3$ is then given as the normal matrix-vector product

$$x' = T \cdot x = T \begin{pmatrix} x \\ 1 \end{pmatrix} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix} = \begin{pmatrix} Rx + t \\ 1 \end{pmatrix}. \tag{21}$$

Concatenation is given by the ordinary 4-dim matrix-matrix product.

The inverse transform is

$$T^{-1} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} R^{-1} & -R^{-1}t \\ 0 & 1 \end{pmatrix} \tag{22}$$

Translation and quaternion: A transformation can efficiently be stored as a pair (t, r) of a translation vector t and a rotation quaternion r. Analogous to the above, the application of (t, r) on a vector x is $x' = t + r \cdot x$; the inverse is $(t, r)^{-1} = (-r^{-1} \cdot t, r^{-1})$; the concatenation is $(t_1, r_1) \circ (t_2, r_2) = (t_1 + r_1 \cdot t_2, r_1 \circ r_2)$.

2.2 Dynamic transformations

Just as static transformations map between (static) coordinate frames, dynamic transformations map between moving (inertial) frames which are, e.g., attached to moving bodies. A dynamic transformation is described by a tuple (t, r, v, w) with translation t, rotation r, velocity v and angular velocity w. Under a dynamic transform (t, r, v, w) a position and velocity $(x, \dot{x})$ maps to a new position and velocity $(x', \dot{x}')$ given as

$$x' = t + r \cdot x \tag{23}$$

$$\dot{x}' = v + w \times (r \cdot x) + r \cdot \dot{x} \tag{24}$$

(the second term is the additional linear velocity of $\dot{x}'$ arising from the angular velocity w of the dynamic transform). The concatenation $(t, r, v, w) = (t_1, r_1, v_1, w_1) \circ (t_2, r_2, v_2, w_2)$ of two dynamic transforms is given as

$$t = t_1 + r_1 \cdot t_2 \tag{25}$$

$$v = v_1 + w_1 \times (r_1 \cdot t_2) + r_1 \cdot v_2 \tag{26}$$

$$r = r_1 \circ r_2 \tag{27}$$

$$w = w_1 + r_1 \cdot w_2 \tag{28}$$

For completeness, the footnote[1] also describes how accelerations transform, including the case when the transform itself is accelerating. The inverse $(t', r', v', w') = (t, r, v, w)^{-1}$ of a dynamic transform is given as

$$t' = -r^{-1} \cdot t \tag{35}$$

$$r' = r^{-1} \tag{36}$$

$$v' = r^{-1} \cdot (w \times t - v) \tag{37}$$

$$w' = -r^{-1} \cdot w \tag{38}$$

Sequences of transformations by $T_{A \to B}$ we denote the transformation from frame A to frame B. The frames A and B can be thought of coordinate frames (tuples of an offset (in an affine space) and three local orthonormal basis vectors) attached to two bodies A and B. It holds

$$T_{A \to C} = T_{A \to B} \circ T_{B \to C} \tag{39}$$

where $\circ$ is the concatenation described above. Let p be a point (rigorously, in the affine space). We write p^A for the coordinate vector of point p relative to frame A; and p^B for the coordinate vector of point p relative to frame B. It holds

$$p^A = T_{A \to B}\, p^B\,. \tag{40}$$

2.3 A note on affine coordinate frames

Instead of the notation $T_{A \to B}$, other text books often use notations such as T_{AB} or T_B^A. A common question regarding notation $T_{A \to B}$ is the following:

> *The notation $T_{A \to B}$ is confusing, since it transforms coordinates from frame B to frame A. Why not the other way around?*

I think the notation $T_{A \to B}$ is intuitive for the following reasons. The core is to understand that a transformation can be thought of in two ways: as a transformation of the *coordinate frame itself*, and as transformation of the *coordinates relative to a coodrinate frame*. I'll first give a non-formal explanation and later more formal definitions of affine frames and their transformation.

[1] Transformation of accelerations:

$$\dot{v} = \dot{v}_1 + \dot{w}_1 \times (r_1 \cdot t_2) + w_1 \times (w_1 \times (r_1 \cdot t_2)) + 2\, w_1 \times (r_1 \cdot v_2) + r_1 \cdot \dot{v}_2 \tag{29}$$

$$\dot{w} = \dot{w}_1 + w_1 \times (r_1 \cdot w_2) + r_1 \cdot \dot{w}_2 \tag{30}$$

Used identities: for any vectors a, b, c and rotation r:

$$r \cdot (a \times b) = (r \cdot a) \times (r \cdot b) \tag{31}$$

$$a \times (b \times c) = b(ac) - c(ab) \tag{32}$$

$$\partial_t (r \cdot a) = w \times (r \cdot a) + r \cdot \dot{a} \tag{33}$$

$$\partial_t (w \times a) = \dot{w} \times t + w \times \dot{a} \tag{34}$$

Think of $T_{W\to B}$ as translating and rotating a real rigid body: First, the body is located at the world origin; then the body is moved by a translation t; then the body is rotated (around its own center) as described by R. In that sense, $T_{W\to B} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$ describes the "forward" transformation of the body. Consider that a coordinate frame B is attached to the rigid body and a frame W to the world origin. Given a point p in the world, we can express its coordinates relative to the world, p^W, or relative to the body p^B. You can convince yourself with simple examples that $p^W = T_{W\to B}\, p^B$, that is, $T_{W\to B}$ *also* describes the "backward" transformation of body-relative-coordinates to world-relative-coordinates.

Formally: Let (A, V) be an affine space. A coordinate frame is a tuple $(o, \boldsymbol{e}_1, .., \boldsymbol{e}_n)$ of an origin $o \in A$ and basis vectors $\boldsymbol{e}_i \in V$. Given a point $p \in A$, its coordinates $p_{1:n}$ w.r.t. a coordinate frame $(o, \boldsymbol{e}_1, .., \boldsymbol{e}_n)$ are given implicitly via

$$p = o + \sum_i p_i \boldsymbol{e}_i \ . \tag{41}$$

A transformation $T_{W\to B}$ is a ("forward") transformation of the coordinate frame itself:

$$(o^B, \boldsymbol{e}_1^B, .., \boldsymbol{e}_n^B) = (o^W + t, R\boldsymbol{e}_1^W, .., R\boldsymbol{e}_n^W) \tag{42}$$

where $t \in V$ is the affine translation in A and R the rotation in V. Note that the coordinates $(\boldsymbol{e}_i^B)_{1:n}^W$ of a basis vector $\boldsymbol{e}_i^B$ relative to frame W are the columns of R:

$$\boldsymbol{e}_i^B = \sum_j (\boldsymbol{e}_i^B)_j^W \boldsymbol{e}_j^W = \sum_j R_{ji} \boldsymbol{e}_j^W \tag{43}$$

Given this transformation of the coordinate frame itself, the coordinates transform as follows:

$$p = o^W + \sum_i p_i^W \ \boldsymbol{e}_i^W \tag{44}$$

$$p = o^B + \sum_i p_i^B \ \boldsymbol{e}_i^B \tag{45}$$

$$= o^W + t + \sum_i p_i^B \ (R\boldsymbol{e}_i^W) \tag{46}$$

$$= o^W + \sum_i t_i^W \ e_i^W + \sum_j p_j^B \ (R\boldsymbol{e}_j^W) \tag{47}$$

$$= o^W + \sum_i t_i^W \ e_i^W + \sum_j p_j^B \ (\sum_i R_{ij} \ \boldsymbol{e}_i^W) \tag{48}$$

$$= o^W + \sum_i \Big[t_i^W + \sum_j R_{ij} \ p_j^B \Big] \ e_i^W \tag{49}$$

$$\Rightarrow p_i^W = t_i^W + \sum_j R_{ij} \ p_j^B \ . \tag{50}$$

Another way to express this formally: $T_{W\to B}$ maps *co-variant* vectors (including "basis vectors") forward, but *contra-variant* vectors (including "coordinate vectors") backward; see Wikipedia "Covariance and contravariance of vectors".

3 Kinematic chains

In this section we only consider static transformations compost of translation and rotation. But given the general concatenation rules above, everything said here generealizes directly to the dynamic case.

3.1 Rigid and actuated transforms

A actuated kinematic chain with n joints is a series of transformations of the form

$$T_{W\to 1} \circ Q_1 \circ T_{1\to 2} \circ Q_2 \circ T_{2\to 3} \circ Q_3 \circ \cdots \tag{51}$$

Each $T_{i\text{-}1\to i}$ describes so-called "links" (or bones) of the kinematic chain: the rigid (non-actuated) transformation from the i-1th joint to the ith joint. The first $T_{W\to 1}$ is the transformation from *world* coordinates to the first joint.

Each Q_i is the actuated transformation of the joint – usually simply a rotation around the joint's x-axis with a specific angle, the so-called *joint angle*. These joint angles (and therefore each Q_i) are actuated and may change over time.

When we control the robot we essentially tell it to actuate its joint so as to change the joint angles. There are two fundamental computations necessary for control:

1. For a given n-dimensional vector $q \in \mathbb{R}^n$ of joint angles, compute the absolute frames $T_{W\to i}$ (*world* to link transformation) of each link i.
2. For a given n-dimensional vector $\dot{q} \in \mathbb{R}^n$ of joint angle velocities, compute absolute (*world*-relative) velocity and angular velocity of the ith link.

The first problem is solved by "forward chaining" the transformations: we can compute the absolute transforms $T_{W\to i}$ (i.e., the transformation from *world* to the ith link) for each link, namely:

$$T_{W\to i} = T_{W\to i\text{-}1} \circ Q_i \circ T_{i\text{-}1\to i} \ . \tag{52}$$

Iterating this for $i = 2, .., n$ we get positions and orientations $T_{W\to i}$ of all links in world coordinates.

The second problem is addressed in the next section.

3.2 Jacobian & Hessian

Assume we have computed the absolute position and orientation of each link in an actuated kinematic chain. Then we want to know how a point p_i^W attached to the ith frame (and coordinates expressed w.r.t. the world frame W) changes when rotating the jth joint. Equally we want to know how some arbitrary vector a_i^W attached to the ith frame (coordinates relative to the world frame W) rotates when rotating the jth joint. Let a_j^W be the unit length rotation axis of the jth joint (which, by convention, is $r_{W\to j} \cdot (1, 0, 0)$ if $r_{W\to j}$ is the rotation in $T_{W\to j}$). In the following we drop the superscript W because all coordinates are expressed in the world frame. Let q_j be the

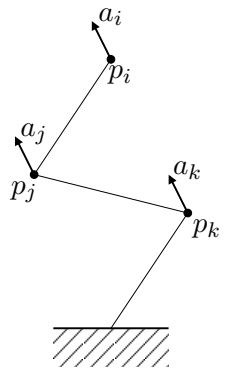

Figure 1: Illustration for the Jacobian and Hessian.

joint angle of the jth joint. For the purpose of computing a partial deriviative of something attached to link i w.r.t. the actuation at joint j we can think of everything inbetween as rigid. Hence, the point and vector Jacobian and Hessian are simply:

$$d_{ij} := p_i - p_j$$

$$\frac{\partial p_i}{\partial q_j} = a_j \times d_{ij} \tag{53}$$

$$\frac{\partial a_i}{\partial q_j} = \alpha_j \times a_i \tag{54}$$

$$\begin{aligned}\frac{\partial^2 p_i}{\partial q_j\, \partial q_k} &= \frac{\partial a_j}{\partial q_k} \times d_{ij} + a_j \times \frac{\partial d_{ij}}{\partial q_k} \\ &= (a_k \times a_j) \times d_{ij} + a_j \times [a_k \times (p_i - p_k) - a_k \times (p_j - p_k)] \\ &= (a_k \times a_j) \times d_{ij} + a_j \times (a_k \times d_{ij}) \\ &\Big[\text{using } a\times(b\times c) + b\times(c\times a) + c\times(a\times b) = 0\Big] \\ &= a_k \times (a_j \times d_{ij}) \end{aligned} \tag{55}$$

$$\begin{aligned}\frac{\partial^2 a_i}{\partial q_j\, \partial q_k} &= (a_k \times \alpha_j) \times a_i + \alpha_j \times (a_k \times a_i) \\ &= a_k \times (\alpha_j \times a_i) \end{aligned} \tag{56}$$

Efficient computation: Assume we have an articulated kinematic *tree* of links and joints. We write $j < i$ if joint j is inward from link (or joint) i. Then, for each body i consider all inward edges $j < i$ and further inward edges $k \le j$ and compute $H_{ijk} = \partial_{\theta_j} \partial_{\theta_k} p_i$. Note that H_{ijk} is symmetric in j and k – so computing for $k \le j$ and copying the rest is sufficient.

4 A note on partial derivatives w.r.t. a vector

Let $x \in \mathbb{R}^n$. Consider a scalar function $f(x) \in \mathbb{R}$. What exactly is the partial derivative $\frac{\partial}{\partial x} f(x)$? We can write it as follows:

$$\frac{\partial}{\partial x} f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \dots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}^\top \tag{57}$$

Why did we write this as a transposed vector? By definition, a derivative is something that, when you multiply it with an infinitesimal displacement $\delta x \in \mathbb{R}^n$, it returns the infinitesimal change $\delta f(x) \in \mathbb{R}$:

$$\delta f(x) = \frac{\partial f(x)}{\partial x}\, \delta x\ . \tag{58}$$

Since in our case $\delta x \in \mathbb{R}^n$ is a vector, for this equation to make sense, $\frac{\partial f(x)}{\partial x}$ must be a transposed vector:

$$\delta f(x) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \dots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}^\top \begin{pmatrix} \delta x_1 \\ \delta x_2 \\ \dots \\ \delta x_n \end{pmatrix} \tag{59}$$

$$= \sum_i \frac{\partial f(x)}{\partial x_i}\, \delta x_i\ . \tag{60}$$

Expressed more formally: The partial derivative corresponds to a 1-form (also called dual vector). The parameters of a 1-form actually form a co-variant vector, not a contra-variant vector; which in normal notation corresponds to a transposed vector, not a normal vector. See Wikipedia "Covariance and contravariance of vectors".

Consider a vector-valued mapping $\phi(x) \in \mathbb{R}^d$. What exactly is the partial derivative $\frac{\partial}{\partial x}\phi(x)$? We can write it as follows:

$$\frac{\partial}{\partial x}\phi(x) = \begin{pmatrix} \frac{\partial \phi_1(x)}{\partial x_1} & \frac{\partial \phi_1(x)}{\partial x_2} & \dots & \frac{\partial \phi_1(x)}{\partial x_n} \\ \frac{\partial \phi_2(x)}{\partial x_1} & \frac{\partial \phi_2(x)}{\partial x_2} & \dots & \frac{\partial \phi_2(x)}{\partial x_n} \\ \vdots & & & \vdots \\ \frac{\partial \phi_d(x)}{\partial x_1} & \frac{\partial \phi_d(x)}{\partial x_2} & \dots & \frac{\partial \phi_d(x)}{\partial x_n} \end{pmatrix} \tag{61}$$

$$= \boxed{d \times n} \tag{62}$$

Why is this a $d \times n$ matrix and not the transposed $n \times d$ matrix? Let's check the infinitesimal variations again: We have $\delta x \in \mathbb{R}^n$ and want $\delta\phi \in \mathbb{R}^d$:

$$\delta\phi(x) = \frac{\partial \phi(x)}{\partial x}\, \delta x \tag{63}$$

$$= \begin{pmatrix} \frac{\partial \phi_1(x)}{\partial x_1} & \frac{\partial \phi_1(x)}{\partial x_2} & \dots & \frac{\partial \phi_1(x)}{\partial x_n} \\ \frac{\partial \phi_2(x)}{\partial x_1} & \frac{\partial \phi_2(x)}{\partial x_2} & \dots & \frac{\partial \phi_2(x)}{\partial x_n} \\ \vdots & & & \vdots \\ \frac{\partial \phi_d(x)}{\partial x_1} & \frac{\partial \phi_d(x)}{\partial x_2} & \dots & \frac{\partial \phi_d(x)}{\partial x_n} \end{pmatrix} \begin{pmatrix} \delta x_1 \\ \delta x_2 \\ \dots \\ \delta x_n \end{pmatrix} \tag{64}$$

$$= \boxed{d \times n}\ \boxed{n} \tag{65}$$

$$= \begin{pmatrix} \sum_i \frac{\partial \phi_1(x)}{\partial x_i}\, \delta x_i \\ \sum_i \frac{\partial \phi_2(x)}{\partial x_i}\, \delta x_i \\ \vdots \\ \sum_i \frac{\partial \phi_d(x)}{\partial x_i}\, \delta x_i \end{pmatrix} \tag{66}$$

Loosly speaking: taking a partial derivative always *appends* a (co-variant) index to a tensor. The reason is that the resulting tensor needs to multiply to a (contra-variant infinitesimal variation) vector from the right.