Continuous Optimization Framework for Depth Sensor Viewpoint Selection

Behnam Maleki, Alen Alempijevic and Teresa Vidal-Calleja

January 22, 2019

Continuous Optimization Framework for Depth Sensor Viewpoint Selection

Behnam Maleki, Alen Alempijevic and Teresa Vidal-Calleja

Centre for Autonomous Systems, University of Technology Sydney, 15 Broadway
Ultimo, NSW 2007, Australia,
behnam.maleki@uts.edu.au,

Abstract. Distinguishing differences between areas represented with point cloud data is generally approached by choosing a optimal viewpoint. The most informative view of a scene ultimately enables to have the optimal coverage over distinct points both locally and globally while accounting for the distance to the foci of attention. Measures of surface saliency, related to curvature inconsistency, extenuate differences in shape and are coupled with viewpoint selection approaches. As there is no analytical solution for optimal viewpoint selection, candidate viewpoints are generally discretely sampled and evaluated for information and require (near) exhaustive combinatorial searches. We present a consolidated optimization framework for optimal viewpoint selection with a continuous cost function and analytically derived Jacobian that incorporates view angle, vertex normals and measures of task related surface information relative to viewpoint. We provide a mechanism in the cost function to incorporate sensor attributes such as operating range, field of view and angular resolution. The framework is evaluated as competing favourably with the state-of-the-art approaches to viewpoint selection while significantly reducing the number of viewpoints to be evaluated in the process.

Keywords: Perception, Optimal Viewpoint, Depth-Sensors

1 INTRODUCTION

Humans are inherently capable of determining discriminative viewpoints when perceiving 3D objects; generally the view selected emphasizes distinguishing attributes of an object relative to the task at hand. Assuming that position and orientation of viewer constitute the viewpoint (the same as pose), the definition of *optimal viewpoint* is highly dependent on the concept of information required for a task. Examples include optimal views to discern an object from others in clutter [23], to achieve complete coverage of space [9, 19] and to determine distinctiveness between objects of the same class [6]. Our particular interest is in using a depth sensor as the viewer of objects from same class, (e.g human face, single species body attributes - Angus cattle muscle score [14]), as they exhibit limited variations over the entire class.

When examining a 3D object for discriminating tasks in dynamic scenes (e.g. moving animals), limited number of embedded fixed-pose depth sensors in the scene are often confronted with restricted time and angles for capturing data. Systems that perform dense reconstruction [15] continuously add measurements into an underlying representation, the rapid dynamic nature of animal motion and inability to reasonably constrain animals for any periods of time renders these approaches unfeasible on a scale. However, it is reasonable to obtain dense representative point clouds of exemplars (object, animal or part thereof). Then, the problem of discerning an object can inherently be approached leveraging this information and devising methods to acquire the discriminative properties with a limited number of informative viewpoints (in the context of our work referred to as *optimal viewpoints*).

Assigning information content conveyed by each point of a 3D surface is achieved with saliency measures related to local or global curvature [8, 13, 20]. Accordingly, the optimal viewpoint should enable a sensor (depth camera) to have the optimal coverage over the most informative or distinct points both locally and globally while also accounting for the distance to the foci of attention. The viewpoint *quality* and ultimate selection are generally decoupled, candidate viewpoints are discretely sampled on a lattice [12, 13] and evaluated for information.

In this paper we tackle the problem of finding the optimal viewpoint for a depth sensor by presenting a consolidated framework based on optimization on the manifold. A continuous cost function that incorporates view angle, surface normals and surface quality with an analytically derived Jacobian is leveraged via an optimization framework to determine the optimal viewpoint. We further provide, via a single coefficient embedded in the cost function, a mechanism to consider knowledge of the entire object or limited fields of view (FoV). By ray-tracing the point cloud, the framework enables to consider sensor attributes such as optimal sensing range, field of view and angular resolution. The proposed framework is extendable to configurations of multiple depth sensors.

The organization of this paper is as follows: Section II discusses related work on exploiting surface information, with emphasis on viewpoint selection to discriminate objects. Section III details our approach that incorporates surface and viewpoint quality in a unified optimization framework. An empirical evaluation of the approach is presented and we compare our results with the state-of-the-art in Section IV. Finally, conclusions are drawn and future work proposed in Section V.

2 RELATED WORK

Assessment of a 3D object exploits stereopsis, which supports the perception of a 3D world including discriminating a difference in depth, judging slant or curvature, and ascertaining surface properties. Previous research demonstrated that an observer's perception of convexity and concavity of surfaces, even with partial occlusions, allows to ignore task irrelevant regions and shapes of a 3D surface [5, 16].

In order to define regions of interest, measures of surface saliency have been proposed [20]. Measures of centre surround operators have been used, such as maximum curvature [8] or Gaussian-weighted mean curvatures [11]. Alternatively, local descriptors are used, such as Scale Invariant Features [17] and Spin Images [10]. These methods identify regions where curvature is inconsistent with its immediate surroundings and where human vision tends to be drawn to take differences in shape into account. Additionally, work on grouping regions of interest (ROI) and exemplifying influence of extrema points has also been researched [13]. All of these approaches assign weights per vertex or face of a 3D object based on the computed saliency, thereafter this information is the utility to determine optimal viewpoints.

Irrespective of the method assigning surface saliency attributes, the process of determining the optimal viewpoints involves a search over the space of viewpoint poses. This search has been generally performed by discretely sampling the space [12], commonly with viewpoints on a sphere encompassing the object, where orientation of each viewpoint is aimed towards the centroid of the object.

While techniques to group the points and limit the search space have been introduced [7, 12, 13], a unified framework that exploits a cost function associated with the surface quality in an optimization framework has not been proposed.

We address this with an approach leveraging a cost function that incorporates finite fields of view and spatial resolution of a 3D sensor and surface quality relative to the viewpoint. An analytical Jacobian is used in an optimization approach to guarantee the convergence.

3 METHODOLOGY

Given a noisy point cloud obtained by depth sensors, the aim of this work is to find the optimal viewpoint —3D pose $(\mathbf{R}, \mathbf{t}) \in \mathrm{SE}(3)$, where $\mathbf{t}$ is the translation and $\mathbf{R}$ is an orthonormal vector base, i.e. the rotation matrix— for the sensor by computing the geometrical properties of the object-sensor setup to capture the most informative region(s) for accurate perception tasks. Our approach considers the field of view and the depth range of the camera as part of the optimization problem.

Surface curvature in this work is extended to point clouds, by considering curvature embedded at each vertex (point) of the surface. The *normal curvature* at each vertex on the surface is the curvature of the planar curve created by intersecting the surface at that vertex with the plane spanned by the tangent vector and the surface normal. By rotating the tangent vector around the surface normal (and subsequently varying the normal curvature) the curvature property of the surface (and vertices) is acquired as two distinct extrema values called principle curvatures while their average is *mean curvature* [3]. In this work, in addition to mean curvature, we take advantage of noise associated with position and normals of vertices as the information of interest and utilise them in the framework as vertex weights.

The scalar value of weights associated with each vertex can be used as information metric to drive viewpoint selection. Hence after a pre-processing stage we define a 7-tuple for each vertex of the surface that contains 3D position, surface normal and an information weight as $(\mathbf{p}_j, \mathbf{n}_j, c_j)$ where $\mathbf{p}_j = (x_j, y_j, z_j)$ and $\mathbf{n}_j = (n_{x_j}, n_{y_j}, n_{z_j})$ respectively.

In this work we solely utilize depth information of an RGB-D camera, per manufacturers specifications [1] depth RMSE is related to horizontal offset from principle point. Therefore, the most accurate measurement of depth is achieved when the sensor is gazing perpendicularly to the points of interest. Having this defined, the main axis of the sensor should be co-linear with the vertex normals of ROI. Specifically, given the afore-mentioned tuple and the current depth sensor pose, the proposed framework aims to minimize iteratively the objective function conformed by the angle between the principal axis of the sensor and the surface normals and the vector that joins the vertex and the sensor as shown in Fig. 1. This is done, while applying the FOV and visibility range constraints. The flowchart of our proposed algorithm is demonstrated in Fig. 2. Note that in this work the terms 'point' and 'vertex' are used interchangeably.

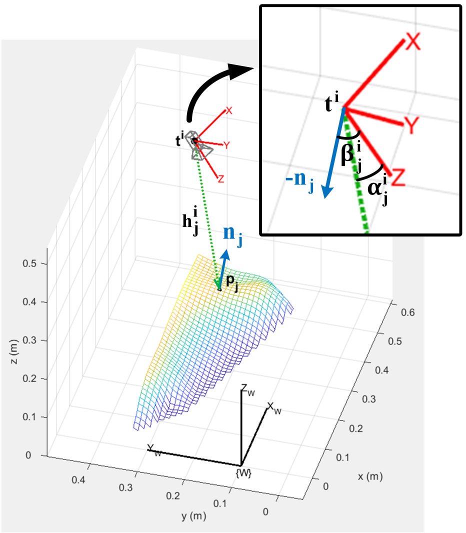


Fig. 1: Angles definition for camera-object setup and magnified area of the camera coordinate frame in the top right. $\mathbf{p}_j$: jth vertex on the surface. $\mathbf{t}^i$ and $\mathbf{R}^i$: camera position and orientation in ith state, respectively. α_j^i: angle between the camera main axis ($\mathbf{z}$ axis of camera) in state i, and vector $\mathbf{h}_j^i(= \mathbf{t}^i - \mathbf{p}_j)$. β_j^i: angle between camera main axis and flipped vertex normal (-$\mathbf{n}_j$). $\left\|\mathbf{h}_j^i\right\|$: distance of camera in ith state from vertex $\mathbf{p}_j$.

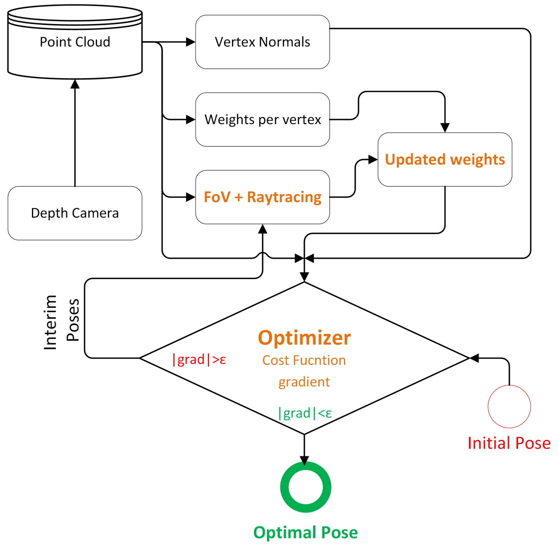


Fig. 2: System Flowchart.

3.1 Angular Terms of Objective Function

Let the camera position $\mathbf{t}$, on the origin of world coordinate system and unit vector of camera axis, be aligned with the Z-axis of the world reference frame. This means that the roll, pitch and yaw components of the camera orientation are zero, which corresponds to a 3×3 identity matrix, i.e. $\mathbf{t}^0 = (0,0,0)$, $\mathbf{R}^0_{3\times3} = \mathcal{I}$. If the camera moves to new pose (position and orientation) of $\mathbf{t}^i$ and $\mathbf{R}^i$, according to SE(3) transformation, the updated orientation of unit vector associated with camera axis is $\mathbf{R}^i\mathbf{z}$.

In Fig. 1, $\mathbf{p}_j$ denotes the coordinate of the j-th point of the point cloud with respect to the world reference frame. By considering the vector $\mathbf{h}^i_j$ formed between the point $\mathbf{p}_j$ and the sensor position $\mathbf{t}^i$, two angles have to be minimized; (1) α: the angle defined between the camera's main axis $\mathbf{R}^i\mathbf{z}$ and $\mathbf{h}^i_j$ and (2) β: angle between the vertex normal, $\mathbf{n}$, and the camera axis, $\mathbf{R}^i\mathbf{z}$. Note that the variable $\mathbf{z}$ represents the unit vector aligned with the Z-axis of reference frame, hence, $\mathbf{z} = [0,0,1]$.

The surface normal at point $\mathbf{p}_j$ is defined by the unit vector $\mathbf{n}_j$. Since the angle between the vertex normal and the camera axis is independent of their position, $\mathbf{n}_j$ can be translated to the camera position. As the minimization procedure includes the angle between $-\mathbf{n}_j$ and the camera axis in terms of the four-quadrant (inverse tangent atan2), the flipped normal surface is used in computations, i.e. $-\mathbf{n}_j$. Thus, α and β are given by:

$$\angle(\mathbf{R}^i\mathbf{z}, \mathbf{t}^i\mathbf{p}_j) : \alpha_j^i = \text{atan2}\, \frac{\left\|\mathbf{R}^i\mathbf{z} \times (\mathbf{p}_j - \mathbf{t}^i)\right\|}{\mathbf{R}^i\mathbf{z} \bullet (\mathbf{p}_j - \mathbf{t}^i)} \tag{1}$$

$$\angle(\mathbf{R}^i\mathbf{z}, \mathbf{n}_j) : \beta_j^i = \text{atan2}\, \frac{\left\|\mathbf{R}^i\mathbf{z} \times -\mathbf{n}_j\right\|}{\mathbf{R}^i\mathbf{z} \bullet -\mathbf{n}_j}\,, \tag{2}$$

where $\|\cdot\|$ is defined as the L2-norm of a vector and $\mathbf{R}^i\mathbf{z}$ is the unit vector defining the camera axis in ith pose. In our computations, α_j^i and β_j^i are confined to the interval $[-\pi, \pi]$.

3.2 Distance Term of Objective Function

The optimal sensing distance at minimum noise of common RGB-D cameras is very limited [18, 1], therefore, the function should include a term to restrict the search of the optimal pose to a ideal distance, η, within the nominal depth sensing range of the sensor (e.g. < 1.0m for Kinect [1, 2]). The distance terms is defined as,

$$h_j^i = \left|\left\|\mathbf{h}_j^i\right\| - \eta\right| = \left|\,\left|\sqrt{(\mathbf{t}^i - \mathbf{p}_j)^\intercal(\mathbf{t}^i - \mathbf{p}_j)}\right| - \eta\right| \tag{3}$$

where h_j^i is a term that aims at setting the distance between the camera's position in ith state, $\mathbf{t}^i$, and the vertex $\mathbf{p}_j$ to the ideal distance, η (see Fig. 1).

3.3 Objective Function

The terms α_j^i and β_j^i are in radians and h_j^i is in meters. Therefore, in order to consolidate them in same order of magnitude, the distance term is multiplied by a coefficient, μ (empirically set, any value between 600-1500 can be selected).

Several different metrics of information c_j, can be attributed to each point (vertex) $\mathbf{p}_j$. For instance c_j's can be defined as mean curvature [3]. If c_{max}, and c_{min} denote the minimum and maximum of the *mean curvature* values, the normalized weight, $w_j \in [0, 1]$, is computed as:

$$w_j = \frac{c_j - \mathrm{c}_{max}}{c_{max} - c_{min}}\,. \tag{4}$$

The weights, w_j's, are not restricted to a specific metric, the noise associated to the vertex positions or vertex normals can also be used (as a proxy we estimate the noise based on the mean of the measurements associated to a vertex).

Finally, a scalar value f integrates the data from all n points as following:

$$\mathrm{f} = \frac{\sum\limits_{j=1}^{n} w_j(\alpha_j + \beta_j + \mu h_j)}{n}\,, \tag{5}$$

Note that the angles and the distance defined above are functions of the state to be estimated R and t:

$$\boldsymbol{\alpha} : SO(3) \times \mathbb{R}^3 \mapsto \mathbb{R} \,:\, (\mathbf{R}, \mathbf{t}) \mapsto \boldsymbol{\alpha}(\mathbf{R}, \mathbf{t}) \tag{6}$$

$$\boldsymbol{\beta} : SO(3) \mapsto \mathbb{R} \,:\, \mathbf{R} \mapsto \boldsymbol{\beta}(\mathbf{R}) \tag{7}$$

$$\mathbf{h} : \mathbb{R}^3 \mapsto \mathbb{R} \,:\, \mathbf{t} \mapsto \boldsymbol{h}(\mathbf{t})\,. \tag{8}$$

3.4 Camera Field of View Constraint

In practice cameras have limited FoV, therefore only some points can be observed at a given position. Moreover, surfaces can be convex therefore some points are obstructed by others. To address these issues, we opted to apply a ray tracing algorithm [21] on the point cloud to find the set of points that are either visible within the FoV, $\mathrm{P}_{_{FoV}}$, or not $\tilde{\mathrm{P}}_{_{FoV}}$.

Given that a full point cloud is available, λ is applied to the weights w_j to allow control over the contribution of points with respect to FoV

$$\begin{cases} w_j \leftarrow \lambda w_j & \text{for } \mathbf{p}_j \in \mathrm{P}_{_{FoV}} \\ w_j \leftarrow (1-\lambda) w_j & \text{for } \mathbf{p}_j \in \tilde{\mathrm{P}}_{_{FoV}} \end{cases} \tag{9}$$

where λ is always equal or greater that 0.5. Note that if λ is equal to 0.5 the FoV is not taken into account, and all points are treated as if they are all visible to the camera.

It is worth mentioning that due to convexity of objects and occlusion the relationship between FoV and the considered points (from the overall pointcloud) is highly non-continuous and non-differential. Therefore we opted to ray trace as a way of sampling as the constraints can not be modelled with an explicit equation.

3.5 Optimization approach

The optimization problem is formulated as:

$$\mathbf{f} : SO(3) \times \mathbb{R}^3 \mapsto \mathbb{R} \,:\, (\mathbf{R}, \mathbf{t}) \mapsto \mathbf{f}(\mathbf{R}, \mathbf{t}) \tag{10}$$

$$(\hat{\mathbf{R}}, \hat{\mathbf{t}}) = \min_{(\mathbf{R}, \mathbf{t}) \in SE(3)} \mathbf{f}(\mathbf{R}, \mathbf{t}) \tag{11}$$

In our proposed implementation, this minimization is carried out by a trust-region solver on Riemannian manifold $\mathcal{M}$. However, to speed up the convergence we derive the analytical Jacobian, $\nabla \mathbf{f}(\mathbf{R}, \mathbf{t}) = (\frac{\partial \mathbf{f}}{\partial \mathbf{R}}, \frac{\partial \mathbf{f}}{\partial \mathbf{t}})$.

In order to compute partial derivative of (1) in vector form with respect to $\mathbf{R}$ and $\mathbf{t}$ we use the auxiliary variable $\mathbf{u_1}(\mathbf{R}, \mathbf{t})$ which is defined as

$$\mathbf{u_1}(\mathbf{R}, \mathbf{t}) = \frac{\|\mathbf{Rz} \times (\mathbf{p} - \mathbf{t})\|}{\mathbf{Rz} \bullet (\mathbf{p} - \mathbf{t})} \tag{12}$$

thus

$$\frac{\partial \boldsymbol{\alpha}}{\partial \mathbf{R}, \mathbf{t}} = \frac{\frac{\partial \mathbf{u}_1}{\partial \mathbf{R}, \mathbf{t}}}{1 + \mathbf{u}_1^2} \tag{13}$$

To compute the $\frac{\partial \mathbf{u_1}}{\partial \mathbf{R}, \mathbf{t}}$, we use the Lemma in the Appendix section and substitute it into the above equation. Similarly for $\boldsymbol{\beta}(\mathbf{R})$, the auxiliary function is defined such that

$$\mathbf{u}_2(\mathbf{R}) = \frac{\|\mathbf{Rz} \times -\mathbf{n}\|}{\mathbf{Rz} \bullet -\mathbf{n}} \tag{14}$$

$$\frac{\partial \boldsymbol{\beta}}{\partial \mathbf{R}} = \frac{\frac{\partial \mathbf{u}_2}{\partial \mathbf{R}}}{1 + \mathbf{u}_2^2} \tag{15}$$

We again use the Lemma in the Appendix section. Note that $\boldsymbol{\beta}$ is function of only $\mathbf{R}$.

As for the distance term

$$\mathbf{h}(\mathbf{t}) = |\ \|\mathbf{t} - \mathbf{p}\| - \eta| \tag{16}$$

using

$$\mathbf{u_3}(\mathbf{t}) = \|\mathbf{t} - \mathbf{p}\| - \eta \tag{17}$$

$$\mathbf{s} = \mathbf{t} - \mathbf{p} \tag{18}$$

$$\mathbf{r} = \|\mathbf{t} - \mathbf{p}\| \tag{19}$$

then,

$$\mathrm{d}\mathbf{t} = \mathrm{d}\mathbf{s} \tag{20}$$

$$\frac{\mathrm{d}\mathbf{h}}{\mathrm{d}\mathbf{t}} = \frac{\mathbf{u}_3}{|\mathbf{u}_3|} \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\mathbf{t}} \tag{21}$$

and finally,

$$\mathbf{r}^2 = \mathbf{s} \cdot \mathbf{s} \tag{22}$$

$$2\mathbf{r}\,\mathrm{d}\mathbf{r} = 2\mathbf{s}\,\mathrm{d}\mathbf{s} \tag{23}$$

And by substituting (24) and (25) into (23) we have:

$$\frac{\mathrm{d}\mathbf{h}}{\mathrm{d}\mathbf{t}} = \frac{\mathbf{u_3}}{|\mathbf{u_3}|} \frac{\mathbf{s}}{\sqrt{\mathbf{s}^\intercal \mathbf{s}}} \tag{24}$$

As the depth sensor pose has 6 DoF, this is an optimization problem on a manifold $SO(3) \times \mathbb{R}^3$. To take advantage of the steepest decent property in trust-regions solver, the computed gradient is projected on the SE(3) manifold, in fact the projected above-computed gradient is the generalization of steepest-descent direction on a Riemannian manifold and absolute value of gradient is used as a metric to stop the optimization. In the case when $\mathbf{u}_3 = 0$, to avoid ambiguity we set the value of $\frac{\mathbf{u_3}}{|\mathbf{u_3}|}$ to 1.

4 EVALUATION

For the evaluation we utilised two point cloud datasets, the cow back (hindquarters) and Igea (venus). We also used three different information measures; mean curvature, vertex normal noise, and vertex position noise.

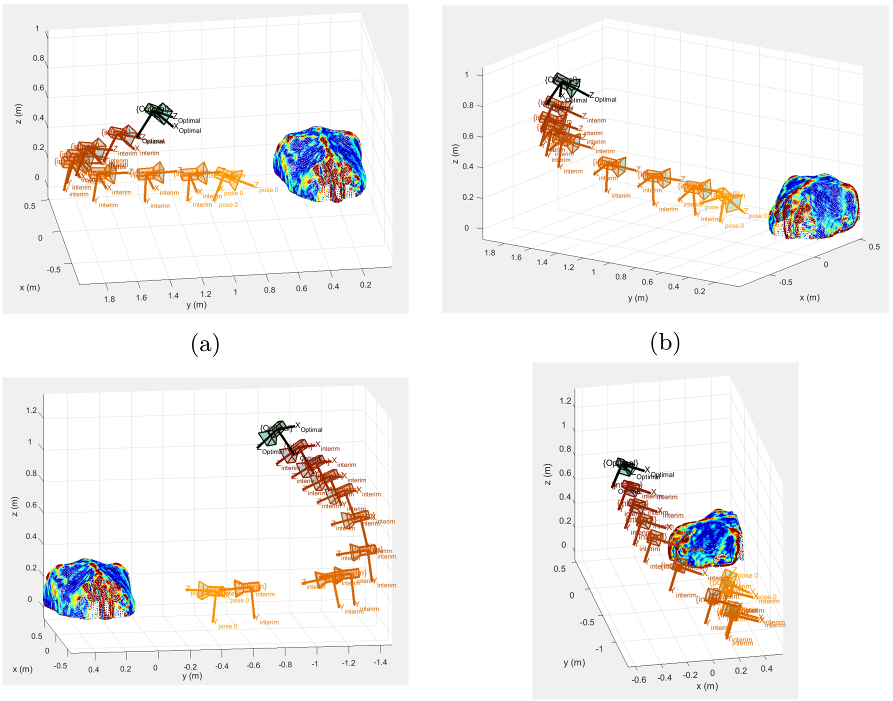
In this work we took advantage of Manopt which is devised for optimization on manifolds [4] and ran the solver of optimization in a system with Core-i5 CPU and 8 Gigabyte RAM. For the first experiment, we performed the optimization on cow back dataset with an initial pose randomly selected from left side of point cloud, using the mean curvature as the information measure. The optimal position, $\mathbf{t}$, and orientation, $\mathbf{R}$ (in terms of its Euler angles, i.e. roll, pitch and yaw) are displayed in the first experiment of Table 1. In this table, the used parameters, namely the optimal distance (η), FoV coefficient (λ), number of iterations, and the elapsed time of entire framework are included. This experiment is illustrated in Fig. 3(a) and (b) from two different viewing angles.

The experiment was also repeated with a random initial pose on the right side of dataset, the result also summarized in Table 1. This experiment is illustrated in Fig. 3(c) and (d) from two different viewing angles. Note that the cow back is almost symmetrical, the two sides are similarly rich in mean curvature information. As depicted in Fig. 3, the two optimal viewpoints (cameras denoted in black) are oriented and positioned similarly with respect to the initial pose. In Fig. 3, color of vertices represents the value of mean curvature with color scale from blue (small mean curvature) to red (large mean curvature). For the purpose of better illustration of the optimization process, the sensor is also colormapped based on its pose from light orange representing the initial guess, through a subset of interim iterations (color scale), and finally to black sensor representing the optimized pose. As explained in section 3.3, the weights w_j (associated with information of interest) in our framework can also be defined based on the noise associated with vertex positions or vertex normals. Several point clouds of Cow Back dataset are used to compute the associated noise. The results and parameters of the optimization utilizing this information are noted

Table 1: Experimental results

Dataset	Weight	η(m)	λ	Ini. Pose	Optimal Pose		Iter.	Time(s)
					$\mathbf{R}$(radian)	$\mathbf{t}$(m)		
Cow B	M. curv.	1.7	0.51	left side	(2.07,0.51,2.71)	[-0.57,1.63,0.85]	90	101
Cow B	M. curv.	1.7	0.51	right side	(-2.27,-0.48,0.53)	[-0.53,-0.94,1.12]	95	110
Cow B	pos. noise.	1.0	0.51	anywhere	(3.01,0.51,1.66)	[-0.21,0.41,1.04]	87	83
Cow B	normal noise	1.0	0.51	anywhere	(2.66,0.66,-0.53)	[-0.39,0.72,0.85]	47	42
Igea	M. curv.	1.4	0.51	front	(-1.06,-0.02,1.41)	[0.29,-0.97,-0.57]	44	36
First point set	M. curv.	1.4	0.51	-	towards centroid	[-0.89,0.16,1.26]	23546	1291
Second lattice	M. curv.	1.4	0.51	-	towards centroid	[-0.95,0.17,1.34]	15000	1147

as third and forth experiments of Table 1 respectively. This experiment is illustrated in Fig. 4(a) and (b) for noise associated with vertex positions or vertex normals, respectively. As the information of interest in these two cases are not symmetrically distributed over the point cloud, the optimal pose is independent of the initial guess.



(a)

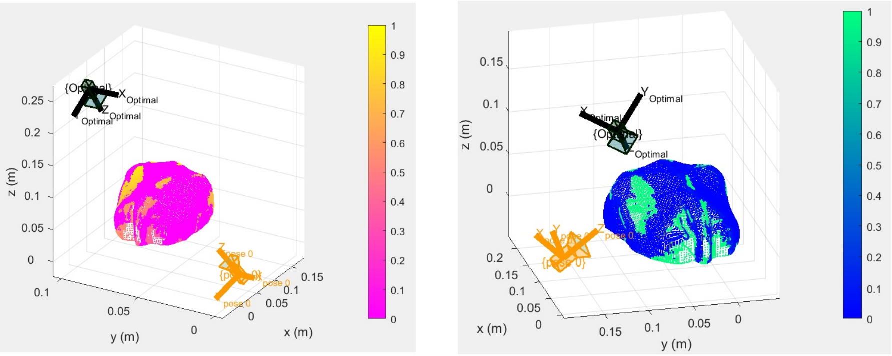
(b)

(c)

(d)

Fig. 3: (a) and (b) two different view angles of one experiment associated with optimization process of camera pose in which the initial guess is randomly selected (pose 0 in light orange) from left side of the cow dataset (color-coded vertices based on mean curvature values) after going through some interim poses (color scale) and ending up to the optimal pose (black camera). (b) and (c) two different view angles of an experiment in which the random initial guess is on the right side of the cow dataset (for the purpose of illustration the scale of camera and the associated axes are magnified).

Our proposed framework performs optimisation in continuous space, we also evaluate against traditional approach of evaluating viewpoints in discrete space. In order to gain an insight into the value of cost function in terms of the position of sensor in discretized space, we constructed two point sets around the cow back dataset to compute the value of cost function **f** per point (of point set)



(a)

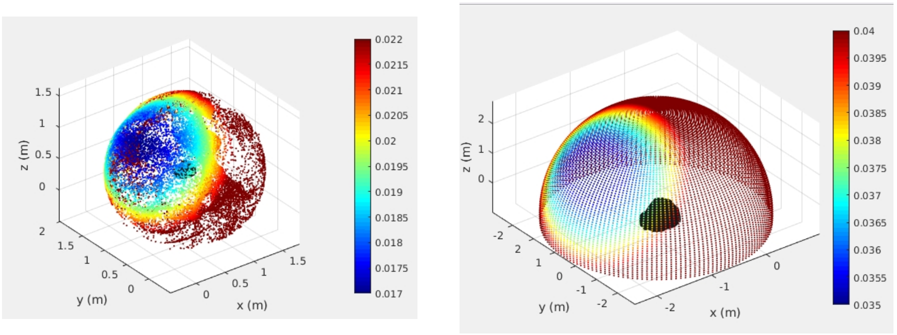
(b)

Fig. 4: (a) Optimal pose resulting from optimization based sparseness (b) Optimal pose resulting from optimization based vertex normal noise. In both figures vertices are color coded with the values of associated normalised noise.

as a single pose with ray tracing (i.e. without optimization). In the first set, Fig. 5(a), two types of candidate points are selected. The first type is in distance of η from the associated vertex in direction of associated normal (the lattice is an expanded version of point cloud in the direction of vertex normals. Here orientation of sensor is in direction of flipped normal of the associated position. To cover all potential poses around the object, the second type is defined as spline interpolated candidate positions using first type oriented towards the centroid of mass.

The second point set, Fig. 5(b), is a lattice of 15000 regularly spaced points over a hemisphere with radius η+0.2 meters (to compensate the distance between centroid and surface) from the centroid of object. The point coordinates of lattice serve as the candidate positions of the depth sensor while oriented towards the centroid of object. For these two point sets the value of cost function is computed and colourmapped per pose. Fig. 5 (a) and (b) display the color scale and the associated values of the cost function computed per pose over the point set, with blue colour indicating poses where value of cost function is smaller (informative viewpoints). The results and parameters of these two experiments are shown in Table 1.

It is imperative to note that the definition of "best" viewpoint is dependent on the weights of interest (i.e. curvature, noise etc.). In the devised experiments for discrete space, the orientation of sensor is constrained towards the flipped normal of vertex (first type) and the centroid of object (second type). Therefore, the naive discrete optimization has 4 DOF (versus 6 DOF in continuous space). Despite this simplification aimed at reducing number of poses to be evaluated for the exhaustive search, according to Table 1 the experiment of discrete optimization on the hardware require 11500 or 12340 seconds for the two point sets (the number of iterations is equal to the number of points). Given different experimental basis with similar conditions, we can conclude from the comparison of discrete and continuous counterparts over Fig. 5 that the position associated with minimum value of objective function is [-1.2,0.15,1.5] is quite close to the corresponding position determined by our proposed continuous optimization. However, our approach clearly outperforms the discrete optimization in terms of processing time. If λ is set to 0.5 (i.e. all the points are in the FoV) the resulting position from continuous optimization is identical to the global minimum of discrete experiments. In the case of $\lambda > 0.5$, continuous optimization is dependent on the initial guess caused by symmetry (mean curvature).



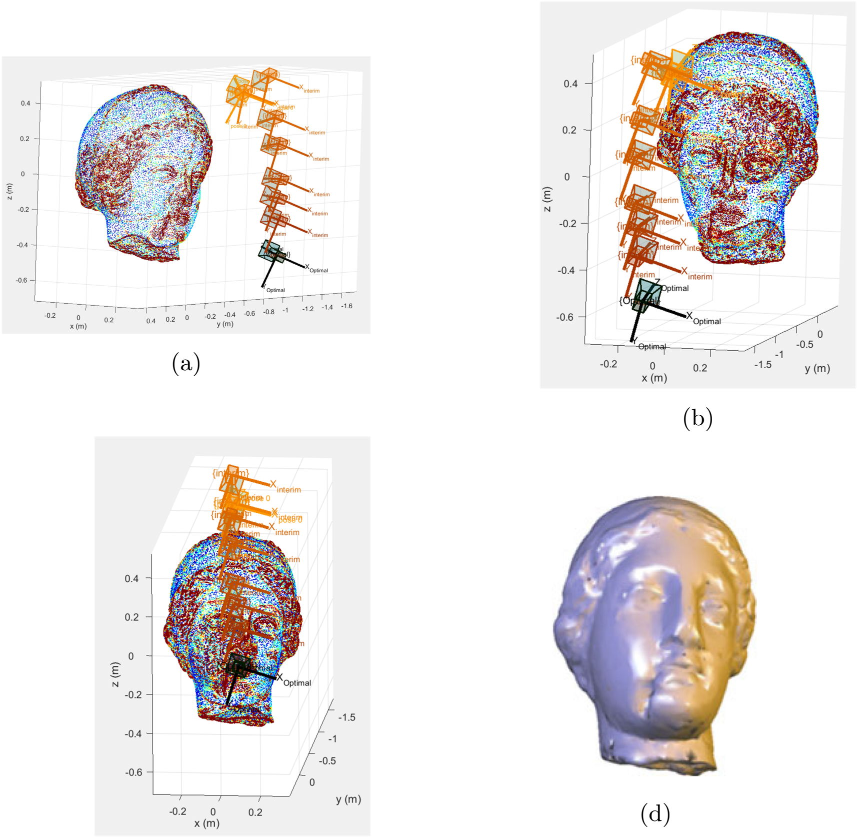
(a)

(b)

Fig. 5: (a) The color map of cost function values associated with the sensor positioned on the first point set (with distance of η =1.4m from vertices in direction of vertex normals) and oriented towards the flipped vertex normal per position. (b) The color map of cost function values associated with the sensor positioned on a hemisphere lattice (with 15000 points with distance of 1.4m from the centroid of object) and oriented towards the centroid of object.

To provide a comparison with a similar discrete sampling approach presented by [13] we tested our framework on Igea (Venus) dataset with parameters stated in the fifth experiment of Table 1. The optimal pose is presented in Table 1 while Fig. 6(a),(b) and (c) depict the optimal pose from three different viewing angles. Similar to the cow back dataset, the mean curvature values are color-coded in vertices and the transition of sensor during the optimization process is indicated with a color scheme from light orange (initial pose) to black (optimal pose) with a subset of the interim states are selected.

Regardless of variation of initial guess (as far as it is not in the back half of head) the optimal pose is unique. The used conditions and assumptions in this experiment are similar to [13] and their optimal viewpoint for this dataset is shown in Fig. 6d. The viewpoint determined by our proposed algorithm is similar to [22] which was used as a benchmark for the work reported by Leifman [13]. The angle displayed by Fig. 6(c) highlights the similarity of optimal viewpoint



(a)

(b)

(c)

(d)

Fig. 6: (a),(b) and (c) three different viewing angles of one optimization experiment (with $\lambda = 0.51$) over Igea dataset (color-coded based on mean curvature values per vertex) after going through some interim poses (shown by color scaled cameras) and ending up to the optimal pose (black camera). (d) optimal viewpoint of Igea illustrated in [13].

rendered by our approach with their result shown in Fig. 6d (for more details refer to [13]). A quantitative compassion over the viewpoint computed was not possible, computer graphics publications such as [22] and [13] only illustrate the viewpoint, not reporting actual values.

The overall performance of our algorithm mainly depends on the variance of information measure on the surface, value of λ, topology of object and the number of vertices. However, the number of iterations through our experiments has proven to be less than 150, which by considering the gradient computation running in parallel with cost function computation, the average computational complexity of our algorithm considering its continuous property is considerably less than discrete approaches.

5 CONCLUSIONS

To address the problem of informative viewpoint of a depth camera, we have shown analytically that by aligning the concept of information with inherent surface characteristics of object surface (such as mean curvature, sparseness, normal noise), the optimal pose is achievable at much lower computational cost compared to numeric approaches. The considered optimal pose of the camera consists of 6 parameters (denoting 6 DoF) and is obtained by minimizing a novel cost function based on geometry of the sensor-object setup through the steepest descent defined by its gradient projected over the appropriate manifold in SE(3). The experiment demonstrates that the optimal pose found by our approach is consistent with the optimal viewpoint of the object obtained via numerical methods and state-of-the-art viewpoint selection approaches.

Future work will incorporate a probabilistic framework to deal with uncertainty of acquiring data from a sensor and update the objective function to continuously optimize the distance of sensor with respect to the surface. We will combine viewpoint selection with machine learning approaches to ascertain the possibility of viewpoint invariance when entire cohorts of animals are evaluated for phenotypic trait estimation purposes. Finally, we aim to integrate our framework for viewpoint selection based on surface quality for inspection of parts developed in additive manufacturing.

Acknowledgments: This work was possible due to the financial and in-kind support, and efforts of many individuals from NSW Department of Primary Industries, University of Technology Sydney, and Meat and Livestock Australia.

References

1. Anders, G.J., Mies, W., N. Sweetser, J., Woodfill, J.: Best-Known-Methods for Tuning Intel RealSense D400 Depth Cameras for Best Performance. Intel (2018)
2. Andersen, M.R., Jensen, T., Lisouski, P., Mortensen, A.K., Hansen, M.K., Gregersen, T., Ahrendt, P.: Kinect depth sensor evaluation for computer vision applications. Technical Report Electronics and Computer Engineering **1**(6) (2012)
3. Botsch, M., Kobbelt, L., Pauly, M., Alliez, P., Lévy, B.: Polygon mesh processing. CRC press (2010)
4. Boumal, N., Mishra, B., Absil, P.A., Sepulchre, R.: Manopt, a matlab toolbox for optimization on manifolds. The Journal of Machine Learning Research **15**(1), 1455–1459 (2014)
5. Cate, A.D., Behrmann, M.: Perceiving parts and shapes from concave surfaces. Attention, Perception, & Psychophysics **72**(1), 153–167 (2010)
6. Chen, X., Saparov, A., Pang, B., Funkhouser, T.: Schelling points on 3d surface meshes. ACM Transactions on Graphics (TOG) **31**(4), 29 (2012)

7. Foissotte, T., Stasse, O., Wieber, P.B., Escande, A., Kheddar, A.: Autonomous 3d object modeling by a humanoid using an optimization-driven next-best-view formulation. International Journal of Humanoid Robotics **7**(03), 407–428 (2010)
8. Gal, R., Cohen-Or, D.: Salient geometric features for partial shape matching and similarity. ACM Transactions on Graphics (TOG) **25**(1), 130–150 (2006)
9. Gonzalez-Barbosa, J.J., García-Ramírez, T., Salas, J., Hurtado-Ramos, J.B., et al.: Optimal camera placement for total coverage. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, pp. 844–848. IEEE (2009)
10. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. IEEE Transactions on pattern analysis and machine intelligence **21**(5), 433–449 (1999)
11. Lee, C.H., Varshney, A., Jacobs, D.W.: Mesh saliency. In: ACM transactions on graphics (TOG), vol. 24, pp. 659–666. ACM (2005)
12. Lee, J., Moghaddam, B., Pfister, H., Machiraju, R.: Finding optimal views for 3d face shape modeling. In: Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on, pp. 31–36. IEEE (2004)
13. Leifman, G., Shtrom, E., Tal, A.: Surface regions of interest for viewpoint selection. IEEE transactions on pattern analysis and machine intelligence **38**(12), 2544–2556 (2016)
14. May, S., Mies, W., Edwards, J., Williams, F., Wise, J., Morgan, J., Savell, J., Cross, H.: Beef carcass composition of slaughter cattle differing in frame size, muscle score, and external fatness. Journal of Animal Science **70**(8), 2431–2445 (1992)
15. Newcombe, R.A., Fox, D., Seitz, S.M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 343–352 (2015)
16. Norman, J.F., Norman, H.F., Lee, Y.L., Stockton, D., Lappin, J.S.: The visual perception of length along intrinsically curved surfaces. Perception & psychophysics **66**(1), 77–88 (2004)
17. Ohbuchi, R., Osada, K., Furuya, T., Banno, T.: Salient local visual features for shape-based 3d model retrieval. In: Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on, pp. 93–102. IEEE (2008)
18. Pece, F., Kautz, J., Weyrich, T.: Three depth-camera technologies compared. In: First BEAMING Workshop, Barcelona, vol. 2011, p. 9 (2011)
19. Quin, P., Paul, G., Alempijevic, A., Liu, D., Dissanayake, G.: Efficient neighbourhood-based information gain approach for exploration of complex 3d environments. In: Robotics and Automation (ICRA), 2013 IEEE International Conference on, pp. 1343–1348. IEEE (2013)
20. Shilane, P., Funkhouser, T.: Distinctive regions of 3d surfaces. ACM Transactions on Graphics (TOG) **26**(2), 7 (2007)
21. Skinner, B., Vidal Calleja, T., Valls Miro, J., De Bruijn, F., Falque, R.: 3d point cloud upsampling for accurate reconstruction of dense 2.5 d thickness maps. In: Australasian Conference on Robotics and Automation (2014)
22. Vieira, T., Bordignon, A., Peixoto, A., Tavares, G., Lopes, H., Velho, L., Lewiner, T.: Learning good views through intelligent galleries. In: Computer Graphics Forum, vol. 28, pp. 717–726. Wiley Online Library (2009)
23. Wu, K., Ranasinghe, R., Dissanayake, G.: Active recognition and pose estimation of household objects in clutter. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on, pp. 4230–4237. IEEE (2015)

6 Appendix

Lemma: Let $\mathbf{t}$, $\mathbf{p}$ and $\mathbf{z}$ be three column vectors in $\mathbb{R}^n$ and $\mathbf{R}$ is a $n \times n$ matrix. Then for $\mathbf{u}(\mathbf{R}, \mathbf{t}) = \frac{\|\mathbf{Rz} \times (\mathbf{p} - \mathbf{t})\|}{\mathbf{Rz} \bullet (\mathbf{p} - \mathbf{t})}$, we have:

$$\frac{\partial \mathbf{u}}{\partial \mathbf{t}} = \frac{(\mathbf{Rz} \cdot \mathbf{Rz})}{((\mathbf{p} - \mathbf{t}) \cdot \mathbf{Rz})^3 \mathbf{u}} \Big((\mathbf{p} - \mathbf{t}) \cdot (\mathbf{p} - \mathbf{t}) \mathbf{Rz} - ((\mathbf{p} - \mathbf{t}) \cdot \mathbf{Rz})(\mathbf{p} - \mathbf{t}) \Big) \tag{25}$$

$$\frac{\partial \mathbf{u}}{\partial \mathbf{R}} = \frac{((\mathbf{p} - \mathbf{t}) \cdot (\mathbf{p} - \mathbf{t}))}{(\mathbf{Rz} \cdot (\mathbf{p} - \mathbf{t}))^3 \mathbf{u}} \Big((\mathbf{Rz} \cdot (\mathbf{p} - \mathbf{t})) \mathbf{Rz} - (\mathbf{Rz} \cdot \mathbf{Rz})(\mathbf{p} - \mathbf{t}) \Big) \mathbf{z}^\mathsf{T} \tag{26}$$

Note: To avoid notational confusion of using parentheses, $\mathbf{u}$ denotes the value of function $\mathbf{u}(\mathbf{R}, \mathbf{t})$.

Proof. Assume $\mathbf{a}$ and $\mathbf{b}$ are two column vectors in $\mathbb{R}^n$. The auxiliary variable $\boldsymbol{\kappa}$ is defined as a scalar function of two vectors $\mathbf{a}, \mathbf{b}$ which first vector, $\mathbf{a}$, is a constant,

$$\boldsymbol{\kappa} = \frac{\|\mathbf{a} \times \mathbf{b}\|^2}{(\mathbf{a} \cdot \mathbf{b})^2} = \frac{(\mathbf{b} \cdot \mathbf{b})(\mathbf{a} \cdot \mathbf{a}) - (\mathbf{b} \cdot \mathbf{a})^2}{(\mathbf{b} \cdot \mathbf{a})^2} = \frac{(\mathbf{b} \cdot \mathbf{b})(\mathbf{a} \cdot \mathbf{a})}{(\mathbf{b} \cdot \mathbf{a})^2} - 1 \tag{27}$$

The differential of $\boldsymbol{\kappa}$ is:

$$\begin{aligned} \mathrm{d}\boldsymbol{\kappa} &= \frac{2(\mathbf{b} \cdot \mathrm{d}\,\mathbf{b})(\mathbf{a} \cdot \mathbf{a})}{(\mathbf{b} \cdot \mathbf{a})^2} - \frac{2(\mathbf{b} \cdot \mathbf{b})(\mathbf{a} \cdot \mathbf{a})(\mathbf{a} \cdot \mathrm{d}\,\mathbf{b})}{(\mathbf{b} \cdot \mathbf{a})^3} \\ &= \frac{2(\mathbf{a} \cdot \mathbf{a})}{(\mathbf{b} \cdot \mathbf{a})^3} \Big((\mathbf{b} \cdot \mathbf{a})\mathbf{b} - (\mathbf{b} \cdot \mathbf{b})\mathbf{a} \Big) \cdot \mathrm{d}\,\mathbf{b}\,. \end{aligned} \tag{28}$$

$$\boldsymbol{\kappa} = \mathbf{u}^2 \implies \mathrm{d}\boldsymbol{\kappa} = 2\mathbf{u}\,\mathrm{d}\mathbf{u} \tag{29}$$

by substituting:

$$\mathbf{a} = \mathbf{Rz} \tag{30}$$

$$\mathbf{b} = (\mathbf{p} - \mathbf{t}) \tag{31}$$

$$\mathrm{d}\mathbf{b} = -\mathrm{d}\mathbf{t} \tag{32}$$

$$\mathrm{d}\mathbf{u} = \frac{\mathrm{d}\boldsymbol{\kappa}}{2\mathbf{u}} = \frac{(\mathbf{a} \cdot \mathbf{a})}{(\mathbf{b} \cdot \mathbf{a})^3 \mathbf{u}} \Big((\mathbf{b} \cdot \mathbf{a})\mathbf{b} - (\mathbf{b} \cdot \mathbf{b})\mathbf{a} \Big) \cdot (-\mathrm{d}\mathbf{t}) \tag{33}$$

$$\frac{\partial \mathbf{u}}{\partial \mathbf{t}} = \frac{(\mathbf{a} \cdot \mathbf{a})}{(\mathbf{b} \cdot \mathbf{a})^3 \mathbf{u}} \Big((\mathbf{b} \cdot \mathbf{b})\mathbf{a} - (\mathbf{b} \cdot \mathbf{a})\mathbf{b} \Big) \tag{34}$$

And also if:

$$\mathbf{a} = (\mathbf{p} - \mathbf{t}) \tag{35}$$

$$\mathbf{b} = \mathbf{Rz} \tag{36}$$

$$\mathrm{d}\mathbf{b} = \mathrm{d}\mathbf{R}\,\mathbf{z} \tag{37}$$

$$\begin{aligned} \mathrm{d}\mathbf{u} &= \frac{(\mathbf{a}\cdot\mathbf{a})}{(\mathbf{b}\cdot\mathbf{a})^3\mathbf{u}}\Big((\mathbf{b}\cdot\mathbf{a})\mathbf{b}-(\mathbf{b}\cdot\mathbf{b})\mathbf{a}\Big)\cdot\mathrm{d}\mathbf{R}\,\mathbf{z} \\ &= \frac{(\mathbf{a}\cdot\mathbf{a})}{(\mathbf{b}\cdot\mathbf{a})^3\mathbf{u}}\Big((\mathbf{b}\cdot\mathbf{a})\mathbf{b}-(\mathbf{b}\cdot\mathbf{b})\mathbf{a}\Big)\mathbf{z}^{\mathsf{T}}\cdot\mathrm{d}\mathbf{R} \end{aligned} \tag{38}$$

and finally:

$$\frac{\partial\mathbf{u}}{\partial\mathbf{R}} = \frac{(\mathbf{a}\cdot\mathbf{a})}{(\mathbf{b}\cdot\mathbf{a})^3\mathbf{u}}\Big((\mathbf{b}\cdot\mathbf{a})\mathbf{b}-(\mathbf{b}\cdot\mathbf{b})\mathbf{a}\Big)\mathbf{z}^{\mathsf{T}} \tag{39}$$