Hybrid Model Convolutional Stage-Positive Definite Metric Operator-Infinity Laplacian Applied to Depth Completion

Vanel Lazcano
Núcleo de Matemática, Física y Estadística
Universidad Mayor
Santiago, Chile
0000-0002-5841-1676

Felipe Calderero
VP of Data @ Ladorian — Nuclio Digital School
Madrid, Spain
felipecalderero@gmail.com

***Abstract*—Depth completion is a fundamental task for many applications such as autonomous vehicles, 3D cinema, 3D reconstruction, and others. Many approaches have been proposed to tackle this problem, from classical models (variational models, morphological models, etc.) to convolutional networks. Hybrid models consider the advantages of the convolutional networks to select features and the generalization capacities of the classical models to extrapolate data. A hybrid model that considers convolutional stages and an interpolator has been proposed in the literature. This model assumes an anisotropic metric g_{ij} and the image domain (Ω), embedding the data in a Riemannian manifold $\mathcal{M} = (g_{ij}, \Omega)$. The interpolation task is performed by solving a degenerated partial differential equation in this manifold $\mathcal{M}$. The proposed metric g_{ij} used in this manifold is fundamental to correctly estimate distances between points in the sparse depth data. In this paper, our contributions are two-fold: first, an empirical evaluation of a metric based on a Positive Definite Operator Metric to compare color pixels applied to the depth completion task, and second, a variation of the infinity Laplacian (also the biased infinity Laplacian), namely unbalanced infinity Laplacian (unbalanced biased infinity Laplacian). Both interpolators include a weight map that balances the contribution of different models in the interpolation process. Experimental results in the KITTI Depth Completion Suite dataset, which is publicly available, show that the use of the Positive Definite Metric Operator performs better than the other two models and also performs better than similar models in this dataset.**

***Index Terms*—Infinity Laplacian, geodesic metric, depth completion**

I. Introduction

Depth completion is crucial for many applications such as 3D reconstruction, 3D cinema, video games, autonomous vehicles, and others. Many techniques have been used to tackle the problem of depth completion, from simple interpolation to more sophisticated techniques such as variational models, convolution networks, and deep learning. The depth completion problem starts with a sparse depth image acquired either by a sensor such as a Kinect sensor, Time of Flight camera, LiDar sensor, or estimated by a stereo algorithm. In general, the interpolation of this data follows one of two principal strategies: strategies that use a reference color image to guide the interpolation and procedures that do not use the reference color image.

Our proposal is devoted to models that use the reference color image. We embedded the data in a manifold constructed with the image domain $\Omega \subset \mathbb{R}^2$ and an anisotropic metric g_{ij}. We solved a degenerated partial differential equation in the Manifold (Ω, g_{ij}) to interpolate the depth data. The anisotropic metric contains spatial and photometric terms to compare the similarity between two pixels in the manifold.

In this work, we used a variation of the infinity Laplacian or AMLE model (Absolutely Minimizing Lipschitz Extensions), namely the unbalanced AMLE model, to interpolate depth maps. The AMLE model was discovered by Aronsson in the sixties [1] [2] and revisited in [3]. In [3] AMLE was presented as the simplest interpolator that holds a set of axioms.

The AMLE and the biased AMLE model (bAMLE) were already presented in detail in [4], [5], [6], [7], [8]. To avoid being redundant, we will not present in this manuscript the details of the AMLE and bAMLE implementation; on the other hand, we will explain in detail our proposal, the unbalanced AMLE, and the unbalanced bAMLE.

A. Related works

In [6] authors presented a practical implementation of the AMLE and bAMLE model using different metrics. $\mathcal{L}^1$, $\mathcal{L}^2$, and $\mathcal{L}^{\frac{1}{2}}$ metric. The authors evaluated the model using different color spaces: RGB, XYZ, CIE-Lab, and CMY. The authors also optimized the model's parameters by comparing Particle Swarm Optimization and Elephant Herd Optimization. Their findings were applied to depth completion and optical flow completion. Given the metric and the domain, the authors constructed a Manifold. Varying the exponent of the metric is a way to estimate the shape of the Manifold. A more flexible metric is needed to obtain a better estimation of the shape of the Manifold.

In [9] authors present a non-deep learning model. They filter the available depth data eliminating misaligned points. Afterward, they segment the image into superpixels. If a superpixel is not well represented by a plane, they use the

2832-1324/22/$31.00 ©2022 IEEE
DOI 10.1109/CANDARW57323.2022.00085

convex hull of a series of most inline points. Finally, a pin hole camera model is used to interpolate the filtered data and the remaining data. The KITTI Depth Completion Suite results show that the proposed model performs better than other models based on classical ideas such as variational models or morphological operations.

Nowadays, spatial propagation networks are affinity-based methods used for depth completion, but the model in [10] suffers from the low representation of a fixed affinity and over smoothing. The standard approach estimates independent affinity matrices but in an over-parameterized way. In their work, the authors introduce an efficient model that learns the affinity in neighboring pixels with an attention-based dynamic approach. They use a non-linear propagation model, attention maps, and diffusion suppression. The model requires less iteration to converge, avoiding over-smoothing of the solution and reaching better results than traditional spatial propagation networks.

In [11] authors propose a piecewise depth completion model. The model segments color images into superpixels corresponding to the regions with similar depth values. These superpixels correspond to the same objects and are gathered using a cost map. Obtained results are comparable to the state-of-the-art models. In their evaluation, the authors show the influence of the individual proposed processing stages and the overall performance on the KITTI dataset.

Recently, AMLE and bAMLE model has been used to fill in holes in optical flow or to interpolate elevation models.

II. AMLE Model

Let us consider $I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ a color reference image, an incomplete depth map $u : \Omega \rightarrow \mathbb{R}$ and that the sparse data is located in $O \subset \Omega$ with a boundary ∂O. We endowed the domain Ω with an anisotropic metric g_{ij}, and we constructed a Riemannian manifold $\mathcal{M} = (\Omega, g_{ij})$. Given the manifold, we solve the problem,

$$\Delta_{\infty,g} u = 0 \in \Omega, \tag{1}$$

where u is the interpolated depth data and $u|_{\partial O} = \theta$ the boundary condition represents the available data.

The biased version of the Laplacian or bAMLE is given by,

$$\Delta_{\infty,g} u + \beta |\nabla u|_{\xi} = 0 \in \Omega, \tag{2}$$

where $\beta \in \mathbb{R}$. In equation (2) when $\beta = 0$ we got the AMLE.

A. Considered metric

Considering the discrete domain Ω as a graph, we take two points $\mathbf{x}$ and $\mathbf{y}$ in the grid and its distance should be:

$$d_{\mathbf{xy}} = (\kappa_x \|\mathbf{x} - \mathbf{y}\|^p + \kappa_c \|I(\mathbf{x}) - I(\mathbf{y})\|^q)^r \tag{3}$$

where κ_x, κ_c, p, q and r are positive real values. The estimation of these parameters defines the shape of the manifold. Inspired by the work presented in [12], where symmetric positive definite metric operators are used to estimate the distance between patterns in a feature space, we decided to give more flexibility to the computed distances between points in the discrete grid Ω. The new proposed distance is given by:

$$\begin{aligned} d_{\mathbf{xy}} = (&\kappa_x [(\mathbf{x} - \mathbf{y})^T A (\mathbf{x} - \mathbf{y})]^p \\ &+ \kappa_c [(I(\mathbf{x}) - I(\mathbf{y}))^T C (I(\mathbf{x}) - I(\mathbf{y}))]^q)^r \end{aligned} \tag{4}$$

where A and C are two positive definite matrices whose elements has to be estimated. The measured distante varies with the element values of A and C and is oriented in the features space. This information increases the model's ability to calculate the separation between any two points on the manifold.

As a practical implementation of the distance $d_{\mathbf{xy}}$ we also included the gradient of the image $\nabla I(\mathbf{x})$. Comparison of image gradients instead of intensities or color is a more robust comparison in real scenarios, where the majority of the color variations occur due to illumination changes, shadows, reflections, and other effects,

$$d_{\mathbf{xy}} = (\kappa_x [\gamma(\mathbf{x}, \mathbf{y})]^p + \kappa_c [\delta(\mathbf{x}, \mathbf{y}) + \eta(\mathbf{x}, \mathbf{y})]^q)^r, \tag{5}$$

where $\gamma(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T A (\mathbf{x} - \mathbf{y})$, $\delta(\mathbf{x}, \mathbf{y}) = (I(\mathbf{x}) - I(\mathbf{y}))^T C (I(\mathbf{x}) - I(\mathbf{y}))$, and $\eta(\mathbf{x}, \mathbf{y}) = (\nabla I(\mathbf{x}) - \nabla I(\mathbf{y}))^T B (\nabla I(\mathbf{x}) - \nabla I(\mathbf{y}))$ with A, B and C positive definite symmetric matrices.

B. Structure of A, B and C positive definite matrices

The matrices A, B and C symmetric and positive definite in order to $\mathbf{x}^T A \mathbf{x} > 0$. Let A has the structure,

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix},$$

where a_{11}, a_{12}, and $a_{22} \in \mathbb{R}$. The associated characteristic equation is given by,

$$\lambda^2 - \mathrm{Tr}(A)\lambda + \det(A) = 0.$$

Given that A is symmetric it has real eigenvalues. In order to assure that the eigenvalues of A be positive $a_{11} > 0$ and $\det(A) > 0$ The solution for λ are given by,

$$\lambda = \frac{\mathrm{Tr}(A) \pm \sqrt{\mathrm{Tr}^2(A) - 4\det(A)}}{2}.$$

The discriminant of these equation is positive $\Delta > 0$. In order to assure that the eigenvalues of A be positive, we impose the following condition,

$$\mathrm{Tr}(A) > \sqrt{\mathrm{Tr}^2(A) - 4\det(A)}$$

that is to say, $a_{11}a_{22} - a_{12}^2 > 0$. We selected ramdomly $a_{11} > 0$ and $a_{12} > 0$, thus:

$$a_{22} > \frac{a_{12}^2}{a_{11}}. \tag{6}$$

It means that $\frac{a_1^2 1}{a_1 1}$ is a lower bound of a_{22}. If we select a_{22} as:

$$a_{22} = \frac{a_{12}^2}{a_{11}} + 1.0,$$

the condition in Eq. 6 is satisfied, assuring that A be positive definite, analogously for B and C.

III. Proposed model

We recall that the average gives the numerical implementation of the infinity Laplacian between the positive and the negative eikonal operator,

$$\Delta_{\infty,g}u = \frac{1}{2}\left(\|\nabla u(\mathbf{x})\|_x^+ + \|\nabla u(\mathbf{x})\|_x^-\right) = 0. \tag{7}$$

where $\|\nabla u(\mathbf{x})\|_x^+$ and $\|\nabla u(\mathbf{x})\|_x^-$ are the positive eikonal operator and negative eikonal operator, respectively. We will explain them in the next sub section.

Our proposed model called the unbalanced AMLE o (uAMLE) considers a weight map $\alpha(\mathbf{x}) : \Omega \to [0,1]$ as we show in the following equation,

$$\frac{1}{2}\left(\alpha(\mathbf{x})\|\nabla u(\mathbf{x})\|_x^+ + (1-\alpha(\mathbf{x}))\|\nabla u(\mathbf{x})\|_x^-\right) = 0, \tag{8}$$

this weight map $\alpha(\mathbf{x})$ is a balance term between positive eikonal ($\alpha(\mathbf{x}) = 1.0$), infinity Laplacian ($\alpha(\mathbf{x}) = 0.5$) and also considers the interpolation using negative eikonal operator ($\alpha(\mathbf{x}) = 0.0$). To state this balance term we followed the ideas presented in [14].

The Eq. 8 means that we can have a combination of different interpolation models for every point $\mathbf{x} \in \Omega$. This weight map is defined explicitly as:

$$\alpha(\mathbf{x}) = \frac{1}{1 + e^{\beta_\alpha(|\|\nabla u(\mathbf{x})\|_x^+| - \tau_\alpha|\|\nabla u(\mathbf{x})\|_x^-|)}}, \tag{9}$$

Where β_α and τ_α in $\mathbb{R}$. In one hand, if $|\|\nabla u(\mathbf{x})\|_x^+| \gg \tau_\alpha|\|\nabla u(\mathbf{x})\|_x^-|$ the difference $\|\nabla u(\mathbf{x})\|_x^+| - |\|\nabla u(\mathbf{x})\|_x^-|$ should be positive, the exponential value $e^{\beta_\alpha(|\|\nabla u(\mathbf{x})\|_x^+| - \tau_\alpha|\|\nabla u(\mathbf{x})\|_x^-|)}$ should be large, and the value of the weight map $\alpha(\mathbf{x}) \approx 0$, i.e. the interpolation process is more confident in the positive eikonal operator. In the other hand, if $|\|\nabla u(\mathbf{x})\|_x^+| \ll \tau_\alpha|\|\nabla u(\mathbf{x})\|_x^-|$ the difference $|\|\nabla u(\mathbf{x})\|_x^+| - \tau_\alpha|\|\nabla u(\mathbf{x})\|_x^-|$ should be negative, the exponential value $e^{\beta_\alpha(|\|\nabla u(\mathbf{x})\|_x^+| - \tau_\alpha|\|\nabla u(\mathbf{x})\|_x^+|)}$ should be small, and the value of the weight map $\alpha(\mathbf{x}) \approx 1$, it means that the interpolation process is more confidence in the negative eikonal operator. Finally, if $\alpha(\mathbf{x}) = 0.5$ we recover the AMLE model to interpolate the data.

The same balance term $\alpha(x)$ was applied to the bAMLE, namely the ubAMLE

$$\alpha(\mathbf{x})\Delta_{\infty,g}u + (1-\alpha(\mathbf{x}))\beta|\nabla u|_\xi = 0 \tag{10}$$

if $\alpha(x) = 0.5$ we recover the original version of the bAMLE in Equation 2.

A. Practical model implementation

Given a point $\mathbf{x}$ in the grid and let $\mathcal{N}(\mathbf{x})$ be a neighborhood around $\mathbf{x}$. In [13] is defined the positive eikonal operator:

$$\|\nabla u(\mathbf{x})\|_x^+ = \max_{\zeta\in\mathcal{N}(\mathbf{x})} \frac{u(\zeta) - u(\mathbf{x})}{d_{\mathbf{x}\zeta}}, \tag{11}$$

and the negative eikonal operator,

$$\|\nabla u(\mathbf{x})\|_x^- = \min_{\zeta\in\mathcal{N}(\mathbf{x})} \frac{u(\zeta) - u(\mathbf{x})}{d_{\mathbf{x}\zeta}}. \tag{12}$$

Let $\mathbf{y}$, $\mathbf{z}$ be the location that maximizes the positive eikonal and minimizes the negative eikonal operator, respectively. With this definition, it is possible to state the unbalanced infinity Laplacian,

$$\frac{1}{2}\left(\alpha\left(\frac{u(\mathbf{y}) - u(\mathbf{x})}{d_{\mathbf{xy}}}\right) + (1-\alpha)\left(\frac{u(\mathbf{z}) - u(\mathbf{x})}{d_{\mathbf{xz}}}\right)\right) = 0, \tag{13}$$

where $\alpha = \alpha(\mathbf{x})$.

The solution of equation (13) is given by:

$$u(\mathbf{x}) = \frac{\alpha(\mathbf{x})d_{\mathbf{xz}}u(\mathbf{y}) + (1-\alpha(\mathbf{x}))d_{\mathbf{xy}}u(\mathbf{z})}{\alpha(\mathbf{x})d_{\mathbf{xz}} + (1-\alpha(\mathbf{x}))d_{\mathbf{xy}}}. \tag{14}$$

The iterated version of the unbalanced infinity Laplacian is given by,

$$u^{k+1}(\mathbf{x}) = \frac{\alpha(\mathbf{x})d_{\mathbf{xz}}u^k(\mathbf{y}) + (1-\alpha(\mathbf{x}))d_{\mathbf{xy}}u^k(\mathbf{z})}{\alpha(\mathbf{x})d_{\mathbf{xz}} + (1-\alpha(\mathbf{x}))d_{\mathbf{xy}}} \tag{15}$$

with $k \in \mathbb{N} \cup \{0\}$.

For the ubAMLE we have,

$$u^{k+1}(\mathbf{x}) = \frac{\alpha(\mathbf{x})d_{\mathbf{xz}}u^k(\mathbf{y}) + \alpha(\mathbf{x})d_{\mathbf{xy}}u^k(\mathbf{z}) + a^k(\mathbf{x},\mathbf{y})}{\alpha(\mathbf{x})d_{\mathbf{xz}} + \alpha(\mathbf{x})d_{\mathbf{xy}} - (1-\alpha(\mathbf{x}))\beta d_{\mathbf{xz}}}, \tag{16}$$

where $a^k(\mathbf{x},\mathbf{y}) = \operatorname{sgn}\{u(\mathbf{y}) - u(\mathbf{x})\}(1-\alpha(\mathbf{x}))\beta d_{\mathbf{xz}}u^k(\mathbf{y})$. We observe that the iterative solution of the ubAMLE is very simple. it is weighted average, which is easy and fast to implement.

B. Metric approximation

The geodesic distance between two point $\mathbf{x}$ and $\mathbf{y}$ is given by,

$$L_g(\mathbf{x},\mathbf{y}) = \min\{\text{length}(L)\}, \tag{17}$$

where L is the trajectory that joins $\mathbf{x}$ and $\mathbf{y}$. Practically, we have approximated the length of the geodesic path (L_g) by the distance $d_{\mathbf{xy}}$ defined in Equation 5, i.e.,

$$L_g(\mathbf{x},\mathbf{y}) \approx d_{\mathbf{xy}} \tag{18}$$

in order to simplify the computation of the distance between $\mathbf{x}$ and $\mathbf{y}$. As a proof of concept we have compared experiemntally the geodesic distance with our approximation as in [14]. Let us consider a gray level reference image I in Fig. 1. Given any pair of two pixels x and y, we compute its distance approximated d_{xy}. This approximation let us to reduce the processing time. We present here an experimental comparison with respect the exact d_{xy} given by the Dijkstra algorithm. The computed geodesic distances from the central pixel $\mathbf{x}$ to every

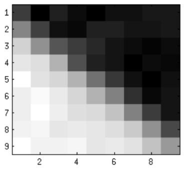
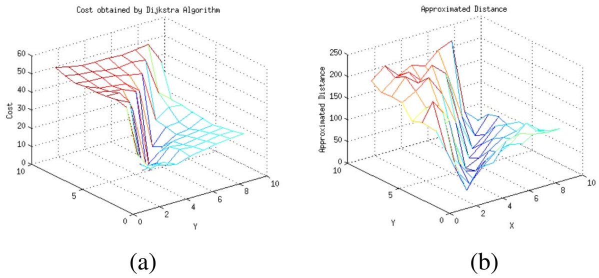


Fig. 1. Squared neighborhoods of the reference image I centered at at pixel $\mathbf{x}$ of radius 4 pixels.



(a)

(b)

Fig. 2. Exact and approximated geodesic distances.

pixel in the neighborhood is displayed in Figure 2, plotted as a surface in Figure 2. The exact geodesic distances are shown in Figure 2 (a) while the approximated distances is shown in (b). Let us notice that the plotted surfaces appear to be similar and we claim that the error committed in the solution of the biased AMLE equation produced by this approximation of the geodesic distance will be small.

IV. Reconfigurable Model

In Figure 3 we show a block diagram explaining the scheme that contains the Convolutional stage-uAMLE model presented in [5]. Figure 3 shows in the top left corner the color reference image. The model processed each color component using Gabor filters, then the model max pool the data in each color component and reconstructed the color reference image (SC1). In the center of the figure, the models use the unbalanced infinity Laplacian (uAMLE), whose inputs are a color reference image and a sparse depth map to complete. The processed color image is used as a reference color image to guide the diffusion process. Then, the uAMLE equation is solved using the model proposed above (see Eq. 15), and their output u is filtered, and max pooled using a final Convolution stage (SC2).

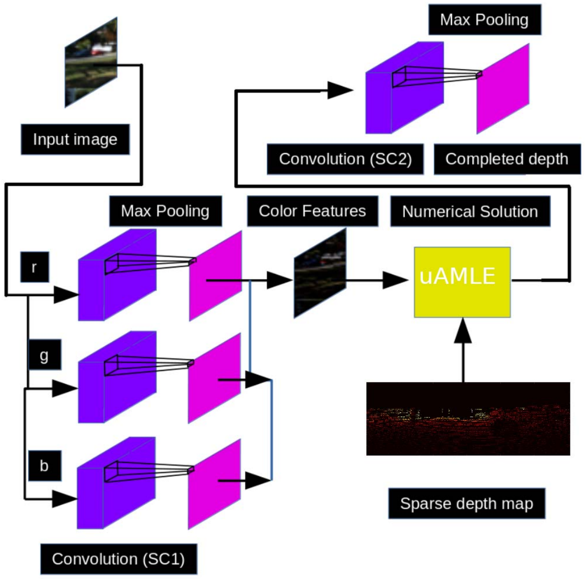
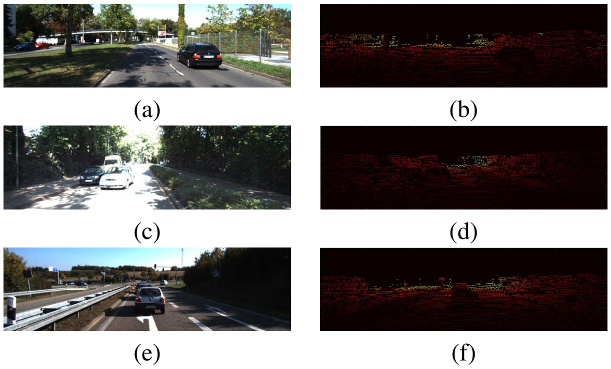


Fig. 3. Reconfigurable scheme for the Convolutional Stage-Infinity Laplacian Model.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 4. Color reference image and its corresponding sparse depth map.

A. Implementation

We have implemented our model in CUDA and MATLAB. The interpolator, the solution of ubAMLE model, was implemented in CUDA 11.5 and gcc compiler 9.4.0 in ubuntu 20.04. The training stage (PSO algorithm) was impelmented with script in MATLAB and exuting the ubAMLE. The software runs in a Laptop i7-8750H in a GPU NVidia Geforce GTX 1060.

V. Training the Model

The proposed model has many parameters that have to be estimated. We present the parameters model in the following table:

TABLE I
Parameters of the Model

Parameter	Description	Number of parameters
κ_c, κ_x, p, q, r	Parameters of the metric	5
$radius$	Neighborhood size of $\mathcal{N}(\mathbf{x})$	1
n_{iter}	iteration number	1
β	bAMLE parameter	1
β_α, β_{tau}	$\alpha(\mathbf{x})$ map parameters.	2
σ_j, ω_j, $j = 1, ..8$	Parameters of Gabor filters.	16
$w1_j$, $w2_j$, $w3_j$, $w4_j$ with $j = 1, ..9$	weights of square average filters	36
a_{11}, a_{12}, b_{11}, b_{12}, c_{11} and c_{12}	Coefficients of A, B and C matrices.	6
Total		68

In Table I we observe that we have 68 parameters of the complete model, where 52 of them are filter parameters.

A. Training set

Taking into account the public available data KITTI Depth Completion Suite in [15], we extracted three color reference images and its corresponding ground truth, as we show in Figure 4.

In Figure 4 (a), (c) and (e) we present three-color reference images used to train the algorithm. In (b), (d), and (f), we present (color-coded) the sparse depth map corresponding to each color reference image. Black color means no available data, and depth values are color-coded between small (red) and large depth values (yellow).

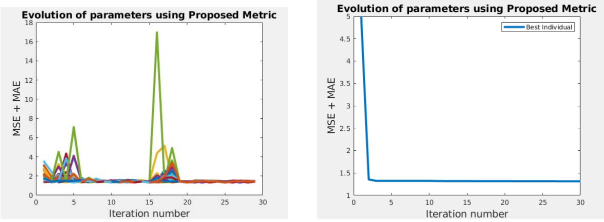
B. Parameter estimation

We defined a vector $\vec{\mu_\mathbf{i}}$ that contains all the 68 parameters. Parameters were estimated using Particle Swarm Optimization (PSO) algorithm. We start with 50 random instances of vector $\vec{\mu_\mathbf{i}}$. Using each vector (parameters), we completed sparse depth maps of the training data set. Thus, we computed a MSE (Mean Square Error) and MAE (Mean absolute value) with the obtained results. The main idea is to minimize a fitness J, which is the addition of the MAE and MSE for each image of the training set,

$$J(\vec{\mu_\mathbf{i}}) = \sum_{k=1}^{\text{Training images Number}} MAE_k + MSE_k \tag{19}$$

In each iteration the PSO algoritm minimizes the fitness J using two evolution equations for each solution μ_i with $i = 1, ..., 50$.

In Figure 5 we show the evolution of the parameter estimation of 50 instances of $\vec{\mu_\mathbf{i}}$. In our cases the training images number is 3.



(a) (b)

Fig. 5. Learning curve for Positive Definite Operator Metric model. (a) Evolution of 50 individuals in 30 iterations. (b) Estimation of best individual.

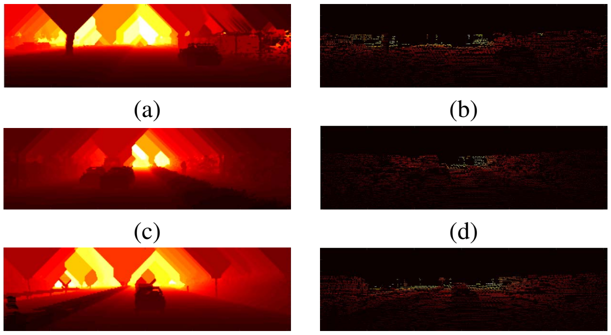
In Figure 5 (a) we show the evolution of 50 individuals in 30 iterations. In (b) we show the evolution of the best individual. We observe that the best individual converges very fast in less that 5 iterations.

In Table II we show the $MSE + MAE$ obtained by three different models we considered in our proposal. We show in Figure 6 the completed depth using the training set and PSO algorithm.

TABLE II
MSE AND MAE IN TRAINING SET

Method	Error
	MSE+MAE
Unbalanced AMLE	1.3149
Unbalanced bAMLE	1.3162
Positive Definite Metric Operator	1.3217

In (b) Figure 6 (b), (d) and (f) sparse depth data and (a), (c) and (e) completed depth data.



(a) (b) (c) (d) (e) (f)

Fig. 6. Completed depth map of the training set. (a) MSE= 1.7153, MAE=0.3951. (b) MSE=0.6788, MAE=0.1989. (c) MSE=0.7622, MAE=0.1946.

C. Parameters of the PSO

We defined the PSO algorithm considering the following elements:

i) Number of individuals: 50 instances.
ii) Stop criteria: 30 iterations.
iii) Fitness of the model: The fitness is presented in Eq. 19.
iv) Evolution of each individual: the position and velocity of each individual is defined according:

$$\vec{\nu_i}^{t+1} = \omega\vec{\nu_i}^{t} + \varphi_g(\vec{\mu_i}^{t} - \vec{\mu_g}) + \varphi_b(\vec{\mu_i}^{t} - \vec{\mu_b}), \tag{20}$$

and,

$$\vec{\mu_i}^{t+1} = \vec{\mu_i}^{t} + \vec{\nu_i}^{t}, \tag{21}$$

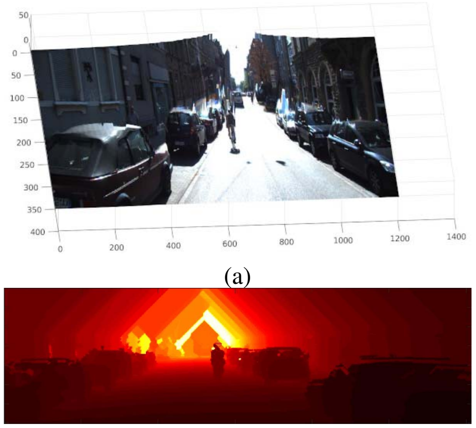
where $\omega = 0.95$ is the evolution parameter for each solution candidate $\vec{\mu_i}$, $\varphi_g = 0.5$ and $\varphi_b = 1.0$ are positive weight parameters, $\vec{\nu_i}$ is the velocity for each candidate solution, $\vec{\mu_b}$ best individual of the current iteration and $\vec{\mu_b}$ is the best individual of all iterations. A saturation for $\vec{\nu_i}$ is usually incorporated to avoid fast change of the solutions. Practically, we used $\nu_{max} = 2$ and $\nu_{min} = -2$.

VI. EXPERIMENTS AND DATASET

We have trained three models we will evaluate in the KITTI data set.

A. Dataset

We used the KITTI Depth Completion Suite dataset available in [15]. KITTI data set contains 1000 images of urban scenes, their corresponding depth map acquired by a LiDar, and their corresponding ground truth. Lidar Scan horizontally the urban scene. The obtained depth map is a sparse image with holes and large areas without information. As shown in Figure 4, the urban scene considers cars, bikes, persons, buildings, trees, and roads presenting illumination changes, blur, reflections, and shadows. As we mentioned above, we selected three reference images and their depth to be used as our training set. The rest 997 images were used to validate our trained model.



(a)

(b)

Fig. 7. Examples of obtained results in the validation set.

B. *Experiments*

We trained and validated three models using KITTI data set. The trained models are:

i) Positive defined Metric Operator. We considered Eq. (15) and we set $\alpha(x) = 0.5$.
ii) Unbalanced AMLE. We solve numerically Eq. (15).
iii) Unbalanced bAMLE. We solve numerically Eq. (16).

VII. Results

Obtained results were evaluated according to the metrics defined in [15]. In Table III we show the obtained results obtained by our models in the validations set. We also present the results obtained by similar model PDC [9] and [8]. We observe that the three of our models outperform PDC. The three models perform similarly, but Positive Definite Operator performs a bit better than the other two models. where *RMSE* is the root mean square error defined as:

TABLE III
RMSE and MAE in Validation set

Method	Error	
	RMSE	***MAE***
PDC [9]	1.2866	0.2932
Lazcano2022 [8]	1.1752	0.3400
Unbalanced AMLE	1.1409	0.3097
Unbalanced bAMLE	1.1422	0.3091
Positive Definite Metric Operator	1.1397	0.3132

$$RMSE = \sqrt{\sum_{i=1}^{N}(gt(i) - u(i))^2}, \tag{22}$$

where gt is the ground-truth, u is the estimated depth and N is the number of points of the image.

In Figure 7 we show an example of the obtained results. In Figure 7 we show in (a) 3D reconstruction and in (b) output of the model.

VIII. Conclusions

We have evaluated three models to interpolate sparse depth data. Our proposal outperforms similar models in the KITTI data set. Specifically, we show that the model AMLE and the Positive definite Metric operator perform better than the uAMLE and ubAMLE. In future work, we will consider improving the geodesic distance approximation between points in the manifold. Also, to extend the evaluation to other data sets.

References

[1] G. Aronsson, “On the partial differential equation $u_x^2u,_{xx} + 2u_xu_yu_{xy} + u_y^2u_{yy} = 0$,” Aktiv für Matematik, vol. 7, 5, pp. 395–425, 1968.

[2] G. Aronsson, Extension of functions satisfyng Lipschitz conditions, Aktiv für Mathematik, vol. 6, 6, pp. 551–561, 1967.

[3] V. Caselles, L. Igual, O. Sander, “An axiomatic approach to scalar data interpolation on surfaces,” Numerische Mathematik, vol. 102, 3, pp. 383–411, 2006.

[4] V. Lazcano, F. Calderero, C. Ballester, “Depth Image Completion Using Anisotropic Operators,” In: , Proceedings of the 12th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2020). SoCPaR 2020. Advances in Intelligent Systems and Computing, vol 1383. Springer, Cham. https://doi.org/10.1007/978-3-030-73689-7_57 (2021).

[5] V. Lazcano, F. Calderero, “Reconfigurable Hybrid Model Convolutional Stage – Infinity Laplacian Applied to Depth Completion,” AICCC ’21: 2021 4th Artificial Intelligence and Cloud Computing Conference December 2021, pp. 108–114 https://doi.org/10.1145/3508259.3508275.

[6] V. Lazcano, F. Calderero, and C. Ballester. “Comparing Different Metrics on an Anisotropic Depth Completion Model”, International Journal of Hybrid Intelligent Systems, Vol. 1, pp: 87 – 99, 10.3233/HIS-210006, 2021.

[7] V. Lazcano, F. Calderero, C. Ballester, “Biased-Infinity Laplacian Applied to Depth Completion Using a Balanced Anisotropic Metric”, In: Liang, Q., Wang, W., Liu, X., Na, Z., Zhang, B. (eds) Communications, Signal Processing, and Systems. CSPS 2021. Lecture Notes in Electrical Engineering, vol 878. Springer, Singapore. https://doi.org/10.1007/978-981-19-0390-8_132, 2022.

[8] V. Lazcano, F. Calderero, “Hybrid Pipeline Infinity Laplacian Plus Convolutional Stage Applied to Depth Completion,” In: Smys, S., Tavares, J.M.R.S., Balas, V.E. (eds) Computational Vision and Bio-Inspired Computing. Advances in Intelligent Systems and Computing, vol 1420. Springer, Singapore. https://doi.org/10.1007/978-981-16-9573-5_8, 2022.

[9] D. Teutscher, P. Mangat, O. Wassermüller, “PDC: Piecewise Depth Completion utilizing Superpixels,” IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, USA, pp. 2752-2758, 2021, doi: 10.1109/ITSC48978.2021.9564656.

[10] Y. Lin, T. Cheng, Q. Zhong, W. Zhou, H. Yang, “Dynamic Spatial Propagation Network for Depth Completion”, AAAI2022 Conference on Artificial Intelligence, 2022.

[11] B. Krauss, G. Schroeder, M. Gustke and A. Hussein, “Deterministic Guided LiDAR Depth Map Completion,” 2021 IEEE Intelligent Vehicles Symposium (IV), 2021, pp. 824-831, doi: 10.1109/IV48863.2021.9575867.

[12] H. Minh, M. San Biagio M, V. Murino, “Log-Hilbert-Schmidt Metric between Positive Definite Operators on Hilbert Spaces,” Proceedings of the 27th Advances in Neural Information Processing Systems, pp. 1-12 ,2014.

[13] J. Manfredi, A. Oberman, A. Svirodov, “Nonlinear elliptic Partial Differential Equations and p-harmonic functions on graphs”, Vol. 28, 12, Differential and Integral Equations, Vol. 28, 2012.

[14] V. Lazcano, “Some problems in depth enhanced video processing”, Ph.D. Thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2016. Available online: http://www.tdx.cat/handle/10803/373917

[15] J. Uhrig ,N. Schneider, L. Schneider, U. Franke, T. Brox, A. Geiger, “Sparsity Invariant CNNs”, International Conference on 3D Vision (3DV), pp 11-20, 2017.