

Self-Organising Maps for Efficient Data Reduction and Visual Optimisation of Stereoscopic based Disparity Maps

Simone Müller
0000-0001-5830-8655
Leibniz Supercomputing Centre (LRZ)
Boltzmannstrasse 1
85748 Garching bei München
simone.mueller@lrz.de

Dieter Kranzlmüller
0000-0002-8319-0123
Ludwig-Maximilians-Universität (LMU)
MNM-Team
Oettingenstr. 67
80538 München
kranzlmue@ifi.lmu.de

Abstract

Many modern autonomous systems use disparity maps for recognition and interpretation of their environment. The depth information of these disparity maps can be utilised for point cloud generation. Real-time and high-quality processing of point clouds is necessary for reliable detection of safety-relevant issues such as barriers or obstacles in road traffic. However, quality characteristics of point clouds are influenced by properties of depth sensors and environmental conditions such as illumination, surface and texture. Quality optimisation and real-time implementation can be resource intensive. Limiting the amount of data allows optimisation of real-time processing. We use Kohonen network existing self-organising maps to identify and segment salient objects in disparity maps. Kohonen networks use unsupervised learning to generate disparity maps abstracted by a small number of vectors instead of all pixels. The combination of object-specific segmentation and reduced pixel number decreases the memory and processing time towards real-time compatibility. Our results show that trained self-organising maps can be applied to disparity maps for improved runtime, reduced data volume and further processing of 3D reconstruction of salient objects.

Keywords

Kohonen Networks, Self-Organising Maps, Depth Image Segmentation, Disparity Maps, Computer Vision

1 INTRODUCTION

Modern fields such as autonomous driving or robotics are computationally intensive and expensive. Real-time based data processing is a mandatory requirement in autonomous vehicles [Mau15]. Spatial image data must be processed in time to detect objects and prevent possible collisions. The depth information of 3D reconstructions in space can be determined by disparity maps of stereoscopic image data. However, the quality of 3D reconstructions is affected by noise, distortion or blurring effects in the images. The use of volumetric data sets can optimise image effects, but real-time processing requires the limitation of processed data volume. The size of volumetric data causes high latency, time delays and additional occurrences of errors.

AI-based computer vision is an promising approach for real-time closed environmental perception. There are sophisticated AI algorithms for the recognition of objects in 2D images, such as YOLOv3 [Red18], Faster Region Based Convolutional Neural Networks (R-CNN) [Ren17] and Multi-Scale Convolutional Neural Network (MSCNN) [Cai16]. However, the previously trained 2D object recognition is limited to concrete object classes and visual interference effects of the images. In order to collect ambient information and make accurate decisions with high confidence, the AI usually needs to be trained extensively with many data sets. Some existing non-trained algorithms detect salient objects based on depth images. These types of algorithms process each pixel of a depth image, which affects the runtime [Hof19]. Even the merging of pixels into colour images is not suitable for a data-reduced and performant application [Ju14].

Our motivation is based on the challenges of AI-data reduction and visual optimisation. By using self-organising maps (SOM), we intend to abstract disparity maps to a reduced number of vectors since a reduced disparity map has a positive impact on computational and memory-based latency. The corresponding SOM algorithm reduces and accelerate the object detection

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

to a few vectors instead of the necessary total pixels. This allows the runtime optimisation.

Our contribution comprises the following aspects:

- Kohonen based depth image segmentation for efficient data reduction
- Concept and implementation of the SOM-algorithm for validation of our approach
- Analysis of data reduction, reliability, runtime and stability of the disparity based SOM-algorithm

Our evaluation reveals the stability and transferability of Kohonen based homogeneity detection of disparity distributions in form of self-organising maps. For this purpose, we use synthetically generated data from the Unreal Engine.

The paper is organised according to a fixed structure consisting of related work, concept and methodology of SOM optimised disparity matching, evaluation, conclusion as well as future work.

2 RELATED WORK

This section describes different related approaches and interrelationships of unsupervised AI technologies for object detection, segmentation as well as performance optimisation.

Class-based Object Detection Object recognition is used across industries as a technology for image-based classification and localisation [Fri22]. Popular AI-algorithms such as YOLOv3, Faster R-CNN and MSCNN use a class-verifying diversity of object propositions for object recognition [Cai16, Mue18, Red18, Ren17]. The number of object proposals can be significantly reduced by using disparity images. General sliding window recognition based methods generate several object proposals of different sizes per sliding window position [Kri19].

Mueller et al. reduce the number of object proposals by creating only one object candidate per sliding window position [Mue18]. The object size is predicted by information such as object dimensions, disparity-based depth information or intrinsic sensor parameters. Most of the detected objects of [Mue18] are additionally upright so that the area of a candidate object has an approximately constant disparity distribution. A homogeneous disparity distribution is collected to remove inappropriate object proposals. The disparity values of all pixels within a candidate object are examined for homogeneity. In case of insufficient disparity homogeneity, the object proposal will not be considered further [Mue18]. The excessive number of object candidates precludes the suitability of this method for disparity based data reduction.

A further possibility for data reduction is the area-reduced creation of disparity map [Pon20]. Objects of

certain classes are identified and localised as input defined stereoscopic images. The MS-CNN algorithm is used for this object recognition [Cai16, Pon20]. The pairwise matching object areas from the content bounding box of the respective stereoscopic images can be determined by a similarity check [Pon20]. The disparity is then determined only for the pixels which represent an object in a matching object area. Limitation resides in the recognition of previously trained object classes.

Superpixel Object Recognition The identification of visually prominent and conspicuous areas are a conducive skill for salient objects recognition [Bor19]. Ju et al. describe an algorithm that processes information from input images and depth maps to create a striking map for detecting salient objects [Ju14]. Thereby, the striking map shows how salient each individual pixel is in comparison to other pixels. Striking objects usually stand out from the surrounding background. Superpixels can be defined by grouping the pixels with same properties like pixel intensity [Jai19]. Ju et al. show that object recognition can be improved with additional depth information in relation to raw colour informations. Salient objects in colour can appear inconspicuous, whereas 3D perceptions are conspicuous. Until now saliency maps have not been optimised for reducing data. Additional steps, such as the creation of a binary map, are necessary to determine relevant sections of the disparity map [Jai19].

Colour and Depth Image Segmentation A division of the depth maps and its colour images into different segments allows the detection of salient objects [Hof19]. Through this approach, each pixel contains additional depth information after segmentation. Fig. 1 describes a search window method to distinguish the object affiliation from the foreground and background.

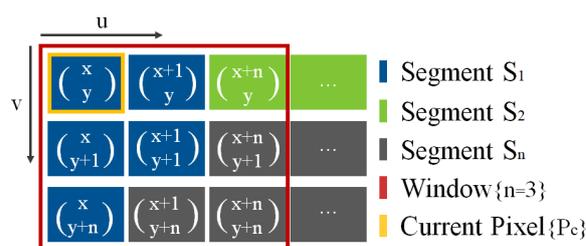


Figure 1: **Square Search Window** [Hof19]: Describes the method for distinguishing object affiliation between the foreground and background. The squares denote the individual segments of S_1 , S_2 and S_n .

Each pixel of the depth map is analysed in a search window for the same segment and the greatest depth. The sum of depth gradients in a segment is subsequently used to allocate foreground or background objects [Hof19]. This methodology leads to a large number of false detections. A considerable number of depth differences is determined on the basis of neighbourhood

observation. The segmentation in [Hof19] is based on the depth image as well as the colour image, but depth differences are still formed for all pixels during object recognition. This affects the running time.

SOM SOM offers promising properties for image segmentation through the use of image features such as colour, texture and pixel intensity [Kau14]. The Kohonen-referred SOM enables the visualisation of high-dimensional data by abstracting original colour sets of matrix-based input data sets [Koh01, Vet18]. The continuous colour set reduction of each pixel allows the following segmentation of the individual areas. Fig. 2 describes the individual processing steps of a SOM.

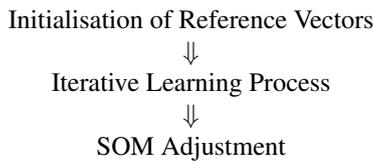


Figure 2: **Pipeline of SOM [Koh01]**

Initialisation of Reference Vectors A classical SOM is represented by a n -dimensional array of neurons. Each neuron μ is connected to a reference vector m_i (Eq. 1).

$$m_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{in}]^T \in R^n \quad (1)$$

The arbitrary selectable values of m_i must be initialised at the start time t_0 . [Koh01]. The reference vectors must have the same dimensions as the input data set. An iterative learning process is carried out after the reference vectors have been initialised.

Iterative Learning Process The input data set shall be defined as a colour-based $[R_n, B_n, G_n]$ input vector $x \in R^n$. From the distance between input vector and all reference vectors, we can determine the neuron that most resembles x . A neuron with the greatest similarity to the input vector x is defined as winner neuron c . The Euclidean distance $d_{E(a,b)}$ is suitable for calculating the distance between the vectors a and b (Eq. 2).

$$d_{E(a,b)} = \sqrt{(\zeta_1 - \eta_1)^2 + (\zeta_n - \eta_n)^2} \quad \begin{cases} a = (\zeta_1, \zeta_n) \\ b = (\eta_1, \eta_n) \end{cases} \quad (2)$$

The direct winner neuron connected reference vector as well as certain reference vectors are adjusted until they are similar to the red coloured input vector x of Fig. 3.

The time-dependent neighbourhood function $h_{ci(t)}$ can be used to determine a neuron in the updating neighbourhood.

$$h_{ci(t)} = \alpha_{(t)} \cdot \exp\left(-\frac{\|r_c - r_i\|^2}{2\sigma_{(t)}^2}\right) \quad (3)$$

$h_{ci(t)}$ defines the learning degree of a corresponding reference vector from the input vector x as shown in Fig. 3.

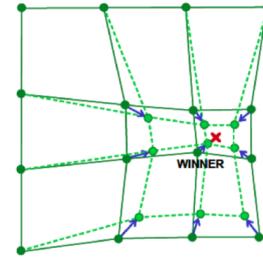


Figure 3: **Determined Winner Neuron [Kah19]:** Update of the selected reference vectors in x -direction

$\alpha_{(t)}$ is described as the learning rate and indicates how strongly the reference vectors should learn from the input vector at time t . The integer value t in Eq. 3 denotes the discrete-time coordinate. By using the neighbourhood radius $\sigma_{(t)}$, the neuron distances to the winning neuron can be determined.

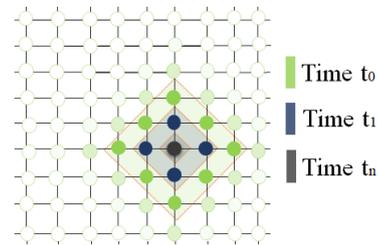


Figure 4: **Neighbourhood Building [Kah19]:** Temporal intensity decrease of the neural neighbourhood. The winner neuron is located in the centre of the coloured neurons.

The functional temporal and monotonic decrease of $\sigma_{(t)}$ causes the equivalent decrease of neighbour neurons around the winning neuron (Fig. 4). Updating the location vector of the winning neuron c ($r_c \in R^n$) and the further neurons i ($r_i \in R^n$) decreases with increasing distance (Fig. 5).

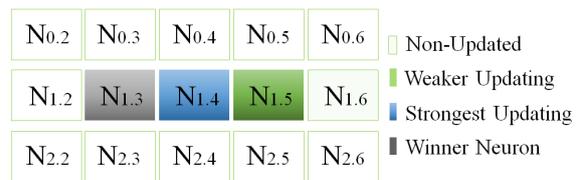


Figure 5: **Distance-Dependent Neuron Updating [Kah19]:** Representation of the location vector based distance influence on the map update. The neuron $N_{1,4}$ is updated more than $N_{1,5}$ due to its closer distance to the winning neuron $N_{1,3}$.

By applying the time-dependent and monotonically decreasing learning rate $\alpha_{(t)}$, we can determine how fast the reference vectors learn from the input vector Eq. 4.

$$m_{i(t+1)} = m_{i(t)} + h_{ci(t)}[x_{(t)} - m_{i(t)}] \quad (4)$$

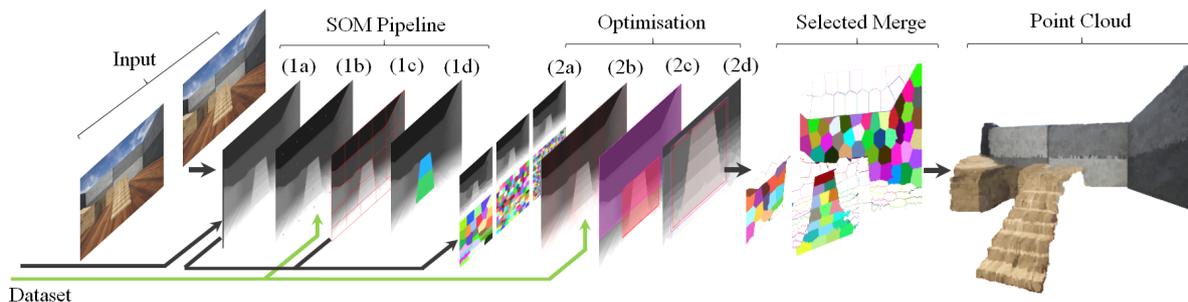


Figure 6: **Pipeline of SOM based Disparity Optimisation:** The pipeline is structured in 4 main steps consisting of actual SOM algorithm, optimisation, merging of object reduced SOM maps and the point cloud rendering.

A special feature of SOM is the dimensionally ordered adoption of reference vectors whereby the reference vectors are similar between neighbouring neurons. Ordering of reference vectors takes place within the initial phase of the iterative learning process. The global order of SOM requires an initial neighbourhood radius of sufficient size and high learning rate values.

The initialisation is followed by the final adjustment of SOM. For this purpose, the learning rate of nearest neighbour neurons should assume small values over a long period of time. The iterative process can be accelerated by initialising already ordered reference vectors [Koh01].

3 KOHONEN NET OPTIMISED DISPARITY MAPPING

In this section we present our SOM based concept for data reduction of disparity maps. Highlighted object areas are identified for this purpose within a disparity map by using kohonen-based SOM. The reduced disparity map has a positive impact on computational and memory-based latency.

Fig. 6 describes the main pipeline of SOM-based data reduction. The steps 1a-1d of the SOM pipeline indicate the following contents (Fig. 7):

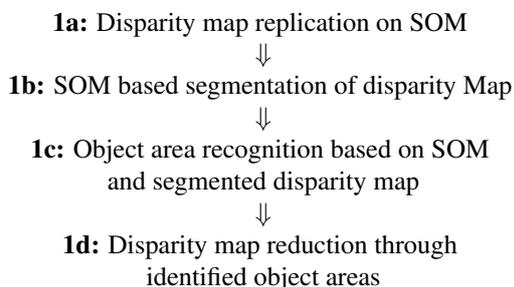


Figure 7: **SOM Pipeline as shown in Fig. 6**

The optimisation of SOM (Fig. 6) occurs through the following parameter and function adjustment:

- SOM dimension

- Input dataset
- Training sequence
- Initial learning rate and neighbourhood radius
- Disintegration function

1a: Disparity Replication Disparity mapping to SOM consists of creating the input dataset, SOM initialisation and training with a used input dataset. The input data set for the SOM algorithm is defined as dimensional matrix (Fig. 8). The original calculation of the

Y-Dimension	[d _[0,0] d _[0,1] d _[0,2] ... d _[0,x-3] d _[0,x-2] d _[0,x-1]]
	[d _[1,0] d _[1,1] d _[1,2] ... d _[1,x-3] d _[1,x-2] d _[1,x-1]]
	[...]
	[d _[y-2,0] d _[y-1,1] d _[y-2,2] ... d _[y-2,x-3] d _[y-2,x-2] d _[y-2,x-1]]
	[d _[y-1,0] d _[y-1,1] d _[y-1,2] ... d _[y-1,x-3] d _[y-1,x-2] d _[y-1,x-1]]
	X-Dimension

Figure 8: **Disparity based Input Dataset**

disparity map may include negative disparity values. To create the input data set, the origin disparity map must be reduced by the negative values.

The step for SOM initialisation includes the definition of map size and initialisation of reference vectors. The dimensions of the reference vectors must match the dimensions of the input data set (y, x, d). Continuing the iterative learning process in the form of a training, we obtain a SOM which might be used for further segmentation and detection of object areas. The training parameters and functions consist of the iteration number, initial learning rate and neighbourhood radius, distance and neighbourhood function as well as the function that defines current learning rate and current neighbourhood radius at time t (Eq. 1, Eq. 2, Eq. 3 and Eq. 4).

The vectors of the input data set can be processed sequentially or in a randomly selected order during training. In case of Fig. 9 we can see the sequentially training result.

A meaningful representation of the disparity map can be achieved by maximising distances between the reference vectors (Fig. 10).

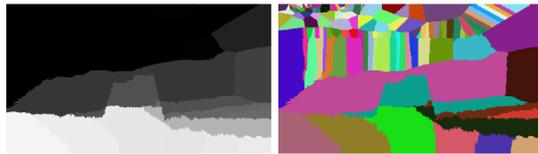


Figure 9: Sequential Training

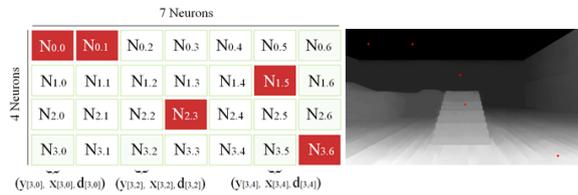


Figure 10: **Selected Reference Vectors in a Disparity Map:** The red dots in the disparity map show the reference vectors. The dimension of the reference vectors consists of $[4 \times 7]$ neurons represented in a data set of (y,x,d) .

1b: Disparity Map Segmentation The number of neurons within the SOM specifies the number of segments in the segmented disparity map. The following steps must occur for each pixel in the segmentation (Fig. 11):

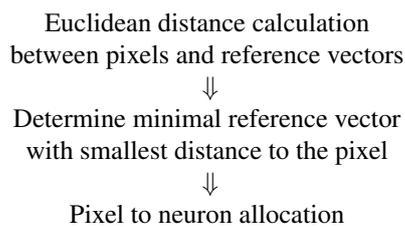


Figure 11: Steps for SOM based Segmentation

All pixels assigned to neurons represent a segment of the disparity map (see Fig. 12 and Fig. 6 in 1c).

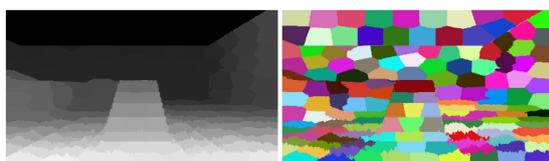


Figure 12: **Segmented Disparity Map:** The left image shows the result of a grey segmented disparity map and right image represents a colour segmented disparity map.

The grey value of the segments is based on the disparity of the associated reference vectors. The segments on the right side of Fig. 12 are coloured with randomly chosen colour values for the visible area delimitation.

The neuron count of SOM is equivalent to the segment count of a disparity map. As Fig. 13 shows, dimension expansion increases the precision of the segmented disparity map.

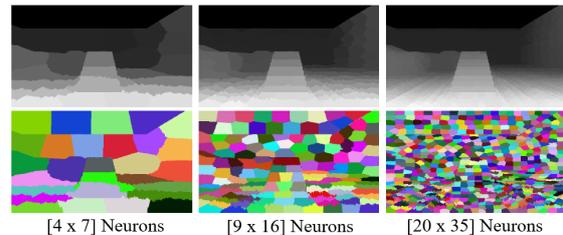


Figure 13: **Neuron-based Dimension Fitting of the Segmented Disparity Maps**

The dimension rise effects the direct growth of running time. Therefore, the requirements must be balanced during the optimisation process.

1c: Disparity based Object Area Recognition The purpose of our object recognition is to identify relevant disparity sections for further data processing. The following pipeline is to be used for this purpose (Fig. 14):

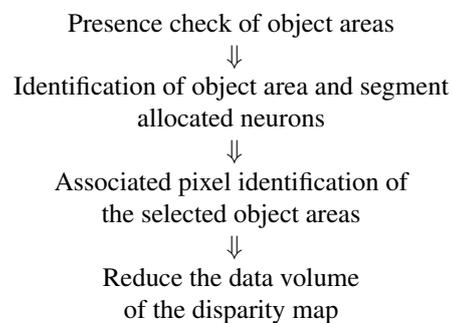


Figure 14: **Object Area Identification for processing based Data Reduction**

An object area can be determined by $k \geq 1$ segments. A strong disparity difference is present when the segment designating neurons are not in close proximity to each other within the SOM. Similar neurons are determined in the neuron set for the object area identification. In case a disturbed neuron occurs between the surrounding neurons, the similar neuron cannot be part of the object area. Fig. 15 denotes the distance influence of similarly identified neurons on the segmented area regions.

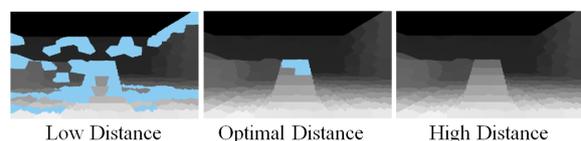


Figure 15: **Neuron Distances dependent Area Recognition**

Disparity based Data Reduction The data volume of a disparity map can be reduced by considering individual object sections. The object selection and the associated pixel reduction leads to a direct runtime reduction. A reasonable tolerance must be implemented, as

not all pixels within a captured object are necessarily taken into account Fig. 16.

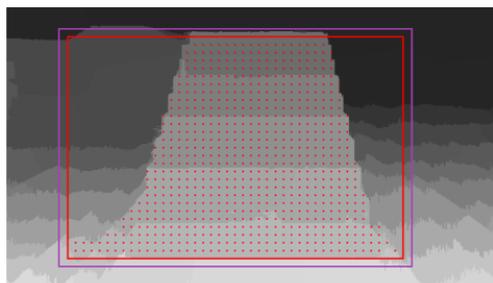


Figure 16: **Trimmed Object Area with and without Tolerance:** The red frame shows the tolerance-free neuron area. The purple frame contains a tolerance outside the stair contour.

4 EVALUATION

Our evaluation employs the SOM algorithm in several measurements to synthetically generated disparity maps. The performance analysis considers designs for runtime-optimised and non-runtime-optimised application cases. The used stair and garden images (Fig. 17 and Fig. 18) are synthetic data from Unreal Engine 4.

Quality Results We examine the number of executions necessary for complete identification of the object areas. Tab. 1 shows that the object areas are not fully identified in every execution.

	100% of Area	95% of Area
NOR_{Stair}	91	91
OR_{Stair}	84	86
NOR_{Garden}	79	91
OR_{Garden}	70	87

Table 1: **Iterations of SOM Algorithm:** Number of exports with compared 100% and 95% recognition of the object areas in 100 executions. The comparison is between the optimised runtime (OR) and the non-optimised runtime (NOR).

The pre-existence of an object area is incorrectly not re-established. In addition, only a partial area of the identified object is recognised. We examined how many executions had at least 95 % of the object areas identified. In areas with many objects and textures, the last 5% could not be fully recognised. Further optimisation of the SOM algorithm can increase the identification rate.

Data Reduction Considering the data reduction, we examine the percentage influence of data volumes with complete identification on average, minimum and maximum reduction.

Tab. 2 shows that the data quantity of the staircase and garden disparity maps is minimally reduced or even increased in the worst case. The unique pink segment in

Fig. 17 shows incorrectly detected areas of other objects. In addition, the data reduction can be lower due to overlaps of relevant area sections (Fig. 18).

	$x_{min}[\%]$	$\bar{x}[\%]$	$x_{max}[\%]$	$x_{opt}[\%]$
NOR_{Stair}	12.56	79.13	86.35	90.04
OR_{Stair}	16.98	75.91	88.11	90.04
NOR_{Garden}	-6.94	39.42	47.81	56.55
OR_{Garden}	7.11	31.26	47.54	56.55

Table 2: **Data Reduction:** Results of the data reduced disparity maps Fig. 17 and Fig. 18 compared between the optimised runtime (OR) as well as the non-optimised runtime (NOR).

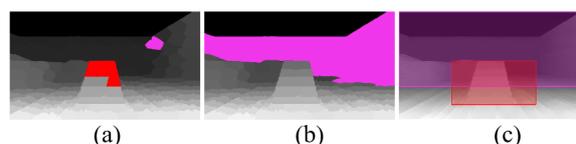


Figure 17: **Segmented Object Area of Stair Image:** Incorrectly identified object area and overlapping object areas. The figure indicates (a) unique segments, (b) faulty object areas and (c) relevant extracts.

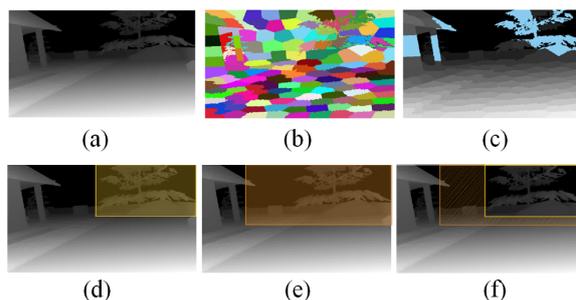


Figure 18: **Identified Object Area of Garden Image:** Description of the large-scale identified object area, which is structured into (a) original disparity map, (b) SOM map, (c) single object segmentation, (d) correct object area, (e) identified object area and (f) excess object area.

Runtime Behaviour The analysis of algorithmic stability behaviour is based on differences in runtime with repeated execution and a constant number of pixels.

A gradual increase from 9072 to 910080 pixels processed on a disparity map involves a maximum linear increase in runtime. Therefore, the number of pixels should be kept as low as possible with adequate result quality. It is noticeable that there is an almost constant runtime behaviour in the recognition of the object areas. Despite the neuron associated object areas, the number of pixels does not correlate with the number of neurons.

Stability Behaviour In the stability analysis, we set the processing pixel count to the runtime-optimised value

of 9072. Our measurements show top and bottom outliers in the detected object areas. The bottom outliers indicate the runs in which no or only very small object areas were detected. The top outliers represent the runs of too large detected object areas. Top and bottom influence leads to short or long runtimes in terms of reduced data volume.

5 CONCLUSION AND FUTURE WORK

In this paper we present a concept for efficient data reduction and visual optimisation of 3D reconstructions by using SOM modified disparity maps. Our motivation is based on the challenges of real-time compatibility and 3D reconstructed quality improvements of visible scattering, distortion, noise and offsets effects. Trained Kohonen networks in form of SOM allow the detection, segmentation and further processed 3D reconstruction of protruding objects from the modified disparity map.

The amount of disparity data can be reduced by the optimised SOM application. Runtime dependency on kohonen training phase requires an optimisation of the SOM algorithm. The direct proportionality between the pixel numbers of a disparity map and the obtained results has effects on processing time. The SOM algorithm proves to be stable with different runtimes and randomness of initial reference vectors and training sequences. Due to the significant influence of randomness, different results can occur during data reduction. One advantage that should be exploited by using SOM is the arrangement of neurons. Neural arrays are suitable for the closer examination of identified object areas. However, a runtime advantage can only be achieved to a limited extent by ordering a SOM. The developed algorithm is particularly suitable for the requirements of a data reduction algorithm.

In our future work we will investigate random dropouts by using non-random sequences of qualitative and constant training results. Additional runtime optimisation will be achieved by parallelising the algorithm at various points. Computational operations in the creation of input data set, segmentation, object recognition, as well as data reduction can be distributed over several processes. We will analyse whether a divide and conquer strategy can be applied to SOM algorithm. For this purpose, the disparity map could be divided into several smaller disparity maps. Protruding detected objects could be reassembled before the data reduction. Further, we will examine the influence of Yig or YCbCr colour models. The recognition features due to striking colours of the Yig or YCbCr could lead to a reduced resolution of disparity map or neuron numbers. Future outsourcing of computational operations to the GPU could prove decently efficient in disparity map segmentation

and training-conditional runtime optimisation. The simultaneous training of several pixels could also lead to promising runtimes. Finally, we will apply the SOM algorithm to real world based depth images.

6 ACKNOWLEDGEMENT

We thank Clara Schönberger for her support during concept development and data collection. We also thank Thomas Odaker, Elisabeth Mayer, Fabio Genz and Daniel Kolb who supported this work with helpful discussions and feedback.

7 REFERENCES

- [Arb11] Arbelaez P., Maire M., Fowlkes C., Malik J., Contour Detection and Hierarchical Image Segmentation. 2011 IEEE Transactions on Pattern Analysis and Machine Intelligence, <https://doi.org/10.1109/TPAMI.2010.161>
- [Cai16] Cai Z., Fan Q., Feris R., A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection., 2016 Computer Vision - ECCV 2016, <https://doi.org/10.48550/arXiv.1607.07155>
- [Bor19] Borji A., Hou Q., Jiang H., Salient object detection: A survey. 2019 Computational Visual Media 5, <https://doi.org/10.1007/s41095-019-0149-9>
- [Fri22] FRITZ LABS INCORPORATED, Object Detection Guide. 2022 Fritz, <https://www.fritz.ai/object-detection/>, last visited April 23th 2022
- [Hof19] Hofmann C., Particke F., Hiller M., Thielecke J., Object Detection, Classification and Localization by Infrastructural Stereo Cameras. 2019 Proceedings of the 14th International Joint Conference on Computer Vision, <https://doi.org/10.3390/electronics9020210>
- [Jai19] Jain D., Superpixels and SLIC. 2019 Medium, <https://darshita1405.medium.com/superpixels-and-slic-6b2d8a6e4f08>, last visited April 23th 2022
- [Ju14] Ju R., Ge L., Geng W., Ren T., Wu G., Depth saliency based on anisotropic center-surround difference. 2014 IEEE International Conference on Image Processing (ICIP), <https://doi.org/10.1109/ICIP.2014.7025222>
- [Kah19] Khazri A., Self Organizing Maps. 2019 IEEE Towards Data Science, <https://towardsdatascience.com/self-organizing-maps-1b7d2a84e065>, last visited April 23th 2022

- [Kau14] Kaur D., Kaur, Y., Various image segmentation techniques: a review. 2014 International Journal of Computer Science and Mobile Computing 3, [https://doi.org/10.1016/0031-3203\(93\)90135-J](https://doi.org/10.1016/0031-3203(93)90135-J)
- [Koh01] Kohonen T., Self-Organizing Maps. 2001 Springer, <https://doi.org/10.1007/978-3-642-56927-2>
- [Kri19] Krigan I., Introduction to Object Detection and Region Proposals. 2019 Brillio Data Science, <https://medium.com/brillio-data-science/object-detection-part-1-introduction-to-object-detection-and-region-proposals-68f6624c98f5>, last visited April 23th 2022
- [Mau15] Maurer M., Autonomes Fahren: Technische, rechtliche und gesellschaftliche Aspekte. 2015 Springer, <https://doi.org/10.1007/978-3-662-45854-9>
- [Mue18] Müller J., Fregin A., Dietmayer K., Disparity Sliding Window: Object Proposals from Disparity Images. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), <https://doi.org/10.1109/IROS.2018.8593390>
- [Pon20] Pon A., Ku J., Li C., Waslander S., Object-Centric Stereo Matching for 3D Object Detection. 2020 IEEE International Conference on Robotics and Automation (ICRA), <https://doi.org/10.1109/ICRA40945.2020.9196660>
- [Red18] Redmon J., Farhadi A., YOLOv3: An Incremental Improvement. 2018 Computer Vision and Pattern Recognition, <https://doi.org/10.48550/arXiv.1804.02767>
- [Ren17] Ren S., He K., Girshick R., Sun J., Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2017 IEEE transactions on pattern analysis and machine intelligence, <https://doi.org/10.1109/TPAMI.2016.2577031>
- [Poz19] Pozo D., Jaramillo K., Ponce D., Torres A., Morales L., 3D reconstruction technologies for using in dangerous environments with lack of light: a comparative analysis. 2019 RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao, <https://www.researchgate.net/publication/334836519>
- [Vac09] Skala V., Cross-talk measurement for 3D displays. 2009 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, <https://doi.org/10.1109/3DTV.2009.5069676>
- [Vet18] Vettigli G., MiniSom: minimalistic and NumPy-based implementation of the Self Organizing Map. <https://github.com/JustGlowing/minisom/>, last visited April 23th 2022