2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

StereoVAE: A lightweight stereo-matching system using embedded GPUs

Qiong Chang[1*], Xiang Li[2], Xin Xu[2], Xin Liu[3], Yun Li[2] and Jun Miyazaki[1]

***Abstract*— We propose a lightweight system for stereo-matching using embedded graphic processing units (GPUs). The proposed system overcomes the trade-off between accuracy and processing speed in stereo matching, thus further improving the matching accuracy while ensuring real-time processing. The basic idea is to construct a tiny neural network based on a variational autoencoder (VAE) to achieve the upscaling and refinement a small size of coarse disparity map. This map is initially generated using a traditional matching method. The proposed hybrid structure maintains the advantage of low computational complexity found in traditional methods. Additionally, it achieves matching accuracy with the help of a neural network. Extensive experiments on the KITTI 2015 benchmark dataset demonstrate that our tiny system exhibits high robustness in improving the accuracy of coarse disparity maps generated by different algorithms, while running in real-time on embedded GPUs.**

I. INTRODUCTION

Stereo-matching is the task of measuring the distance of pixels relative to a camera. The depth information of an object or scene is significantly important in many fields including robotic vision, 3D reconstruction, driver assistance systems etc. It can be extracted from stereo-image pairs using pixel matching along an epipolar line. Although matching methods are more susceptible to a lighting environment than radar and structured-light technologies, they are capable of easily extracting more information (e.g. the shape feature). Thus, they can be efficiently used in practical applications, such as recognition and reconstruction, in a more flexible manner and at a lower cost than conventional technologies.

Recent stereo-matching methods can be divided into the following two categories: traditional methods and learning-based methods. The main difference between these two categories lies in the calculation of the matching costs for each pixel, which is the most critical step in matching. Traditional methods, such as Census and normalized cross-correlation (NCC), usually define a feature template for each pixel to complete the matching. In these methods, the computational cost is low, but only limited accuracy can be achieved. On the other hand, learning-based methods can extract more complex features by training a neural network [1]. This network provides high accuracy but has a high computational cost. Due to their massive advantage regarding the amount of calculations, most current embedded systems often employ traditional methods to estimate the depth information [2] [3]. However, this leads to far less accuracy than that achieved by systems based on general-purpose platforms and often requires auxiliary equipment.

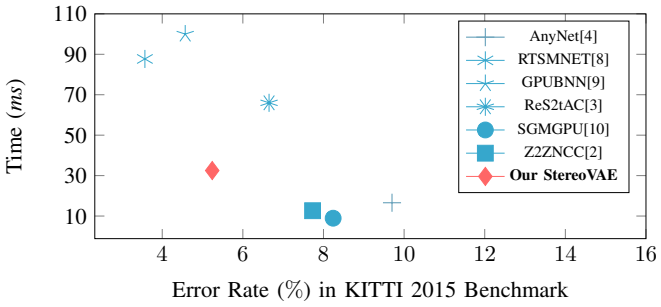

Fig. 1. Accuracy comparison with other systems implemented on a Jetson AGX Xavier GPU.

To achieve improved accuracy, many studies have recently focused on implementing lightweight convolutional neural networks (CNN) for embedded graphic processing units (GPUs) [4]-[7]. However, since most current embedded systems require a significant compromise in terms of processing speed, they often prune their neural networks as much as possible, resulting in limited accuracy. Current CNN-based embedded stereo-vision systems exhibit a limited accuracy improvement over traditional methods in real-time processing [2]. Thus, traditional methods are not far behind in extracting matching features.

Based on the above, we propose a fast stereo-matching frame that combines the advantages of traditional and CNN-based methods as follows:

- we first use the zero-mean normalized cross-correlation (ZNCC) and semi-global matching (SGM) of traditional methods to perform the essential stereo matching in a low-resolution image pair and generate an initial disparity map. Then, we enlarge and optimize this map using a neural network based on a variational autoencoder (VAE);
- we propose a tiny VAE-based super-resolution network (StereoVAE), which employs a quarter size of the initial disparity map as an input and performs the upscaling and refinement;
- we implement our real-time stereo-vision system on a Jetson AGX Xavier GPU and achieve a 5.24% error rate at 30fps by employing the KITTI 2015 dataset.

The proposed hybrid structure can significantly reduce the

[1] Qiong Chang and Jun Miyazaki are with the School of Computing, Tokyo Institute of Technology, Tokyo, 152-8550, Japan. Qiong Chang is the corresponding author. q.chang@c.titech.ac.jp

[2] Xiang Li, Xin Xu and Yun Li are with the School of Electronic Science & Engineering, Nanjing University, Nanjing, 210093, China.

[3] Xin Liu is with the Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, 135-0064, Japan.

running time by exploiting the low computational cost of traditional methods and small images. It then uses the VAE-based neural network, which performs the upscaling and refinement, to compensate for the loss of accuracy. Figure 1 shows the performance comparison of the proposed system with other embedded stereo-vision systems considering the KITTI 2015 benchmark dataset. It is observed that our system achieves the best performance (close to the origin point), low running time (32.5ms) and low error rate (5.24%).

The remainder of this paper is organized as follows. Section II describes the related work. Section III presents the generation methods for a low-resolution disparity image and StereoVAE structure. The experimental results are discussed in section IV. Section V summarizes our work and presents a future research direction.

II. Related Work

This section reviews some fast stereo-vision systems developed using GPUs. All these systems employ neural networks to perform the matching and aim at achieving a high real-time processing speed by employing the KITTI 2015 dataset [11].

Duggal et al. [12] proposed a method for pruning neural networks in a GPU. They employed a CNN method to extract matching features and then a pruning module based on PatchMatch [13] to reduce the amount of calculation using cost aggregation. Zhang et al. [14] proposed a cost aggregation method to reduce the memory and computational cost in a GPU. They added a locally guided aggregation layer, where three K×K filters were employed for each pixel to reduce the loss caused by the downsampling and upsampling layers. Both the above systems achieved high accuracy ($< 3\%$ error rate) by employing the KITTI 2015 dataset but low processing speed even on high-end GPUs (< 20 fps).

Wang et al. [4] proposed a four-stage lightweight neural network based on a Jetson AGX Xavier GPU. First, they obtained only 1/16 of the original size of the disparity map in the initial stage and then propagated the intermediate result to the next stage using a residual network block. This pyramid structure was used to extract the features of different scales and effectively improved the matching accuracy. Their system was the first to achieve error rates of 6.2%–14% at 26–82 fps, making it the fastest learning-based stereo-vision system known in embedded platforms. Chang et al. [6] constructed a pyramid network similar to the AnyNet [4] and introduced a new attention-aware feature aggregation module. This novel module effectively improved the representational capacity of features without the need for excessive additional calculations. Their system was able to perform stereo matching with an 1242×375 image at 12–33 fps on a Jetson Tx2 module and achieved a 7.54% of error rate.

To further improve the matching accuracy, Gan et al. [5] developed a self-adaptive network with an extra convolutional spatial propagation network to refine a coarse disparity map. Using this propagation network, their system reduces the error rate to less than 5% by employing the KITTI 2015 dataset. Furthermore, Dovesi et al. [7] adopted a semantic segmentation structure to further reduce the error rate to 3.3%. However, both the above networks come at a speed price, taking more than 10 ms to process an 1240×375 image.

In addition to the above-mentioned end-to-end systems, the matching accuracy also was improved in [15] to [17] using neural networks to optimize the disparity maps. However, these networks are too large for embedded GPUs with limited resources and cannot meet the high-speed processing requirements of embedded applications.

III. Proposed System

In this section, we introduce the architecture of the proposed system, including 1) the traditional Census and SGM methods for generating low-resolution raw disparity maps and 2) the StereoVAE structure for upscaling and refining the generated low-resolution disparity maps.

A. Coarse Disparity Map Generation

1) Zero-means Normalized Cross-Correlation: ZNCC is a template method that used to apply a matching as follows:

$$C_{ZNCC}(x,y,d) = 1 - \frac{\sum\limits_{(x,y)\in W} \Delta I_R(x,y)\cdot \Delta I_T(x-d,y)}{\sigma_R(x,y)\cdot\sigma_T(x-d,y)}, \tag{1}$$

where

$$\begin{aligned} \sigma_R(x,y) &= \sqrt{\sum_{(x,y)\in W} \Delta I_R(x,y)^2}, \\ \sigma_T(x-d,y) &= \sqrt{\sum_{(x,y)\in W} \Delta I_T(x-d,y)^2}, \end{aligned} \tag{2}$$

and

$$\begin{aligned} \Delta I_R(x,y) &= I_R(x,y) - \overline{I_R(x,y)}, \\ \Delta I_T(x-d,y) &= I_T(x-d,y) - \overline{I_T(x-d,y)}. \end{aligned} \tag{3}$$

Here, the value of $C_{ZNCC}(x,y,d)$ in Eq. 1 represents the similarity between pixels $I_R(x,y)$ and $I_T(x-d,y)$, which is within the range of [0,1]. When the value is small, the similarity is high. In addition, $\sigma_R(x,y)$ and $\sigma_T(x-d,y)$ are the standard deviations of the pixel values in window W and they are used to normalize the correlation coefficient between them.

2) Semi-Global Matching: SGM is one of the most popular optimization methods used for stereo matching [18]. It applies dynamic programming by treating the different path directions equally. The corresponding formula is:

$$\begin{aligned} C_r(x,y,d) =& C(x,y,d) + \min(C_r(x-r,y,d), \\ & C_r(x-r,y,d-1)+P1, \\ & C_r(x-r,y,d+1)+P1, \\ & \min_i C_r(x-r,y,i)+P2) \\ & -\min_k C_r(x-r,y,k). \end{aligned} \tag{4}$$

Here, $C_r(x, y, d)$ represents the optimized cost along the r direction, and $C(x, y, d)$ represents the matching cost between pixels $I_R(x, y)$ and $I_T(x - d, y)$. $P1$ and $P2$ are penalty constants for disparity changes, and i represents the disparity values, except for d and $d \pm 1$. By adding the minimum cost value of the previous pixel with $P1$ and $P2$, the effect of the adjacent pixels is propagated, whereas subtracting the minimum cost value of the previous pixel ensures that the cost value does not overflow. The final cost values $C_{SGM}(x, y, d)$ are then aggregated using different directions as follows:

$$C_{SGM}(x, y, d) = \sum_r C_r(x, y, d) \tag{5}$$

3) WTA and Post-processing: Based on the aggregation results, the disparity map can be generated using

$$D_{map}(x, y) = \arg\min_d (C_{Final}(x, y, d)), \tag{6}$$

where $C_{Final}(x, y, d)$ represents the final matching costs aggregated using Eq.5. Then, we use the single matching phase (SMP) method [19] to remove the occlusion parts and a median filter to reduce the amount of noise. According to the SMP methods, no pixels in the reference image can be matched by two or more pixels in the target image at the same time. Therefore, only the values that meet the following conditions are valid:

$$\forall k \in (0, D) : |x - D_{map}(x, y) - (x - k - D_{map}(x - k, y))| \leq 1. \tag{7}$$

The invalid values are replaced by the closest valid values.

B. Disparity Upscaling and Refinement

1) StereoVAE structure: Figure 2 shows the structure of the proposed StereoVAE. It consists of feature extraction, VAE, and upscaling modules; VAE can be subdivided into encoder and decoder units. Our StereoVAE receives two types of input images: one is the original left image, and the other is the disparity map generated using traditional methods in the previous step. Both are a quarter size of the original image. The output is a high-resolution disparity map amplified by our network. To improve the performance, we also employ skip connections and residual blocks in the network.

Table I shows the structure of our StereoVAE network. X_0 and X_1 represent the low-resolution disparity map and the grayscale left image as feature-extraction (FE) module inputs, respectively. These two inputs are combined to extract the boundary correspondence information between the disparity map and the left image because disparity changes are usually drastic in these regions. The *FE* module reduces the number of features in a dataset by creating new features, which are represented by the existing ones, such as boundary regions. These new features should then be able to summarize most of the information contained in the original dataset. Since disparity maps usually have fewer features than those general images have, our network simply employs two convolutional layers each with 32 5×5 convolutional kernels and a stride of 1 to extract feature information from the low-resolution inputs.

To improve the learning and generalization ability of the network, we introduce a VAE structure. The objective of the VAE module is to reconstruct the features extracted by the *FE* module; the inputs *FE_1* and *FE_2* of the encoder are the outputs of the *FE* module. The number of the outputs of the first layer is 16 with kernels of 5×5 and a stride of 2. In the encoder module, the images are downsampled twice, along with two residual blocks *R* to ensure the backward propagation of features. Each *R* consists of a 5×5 convolutional layer with a stride of 1.

The outputs of the encoder module second layer are downsampled to two dimensions, *mu* and *sigma*, using a convolutional kernel of 1×1. *mu* denotes the mean of a normal distribution, and *sigma* denotes the variance logarithm of a normal distribution. The training target of the VAE is to obtain a standard normal distribution whose mean is 0 and variance is 1. For both *mu* and *sigma*, their training objectives are 0. However, the activation function used in this network is the *leakey_relu*, which has a derivative of $0.1x$ for $x < 0$ and x for $x > 0$, and causes the network to learn at different rates for the positive and negative parts of 0. To eliminate the possible effects due to the difference in the learning speed, we subtract 1 from *mu* and *sigma* to obtain the mean of the normal distribution and the logarithm of the variance, as described in the following equation:

$$gaussian = (mu - 1) + \varepsilon * exp^{(sigma-1)}. \tag{8}$$

Here, *mu* and *sigma* are obtained from the encoder module, ε represents a standard normal distribution, and *Gaussian* is used as the input of the decoder module.

The input of the decoder module is a normal distribution with the mean and variance obtained from the encoder module, which is first increased to 32 dimensions using a deconvolution layer with a 1×1 matrix size and a stride of 1. Output *D_1* is used with *EN_2* and *EN_2_R* to extract feature information using a convolutional layer with a 5×5 kernel. To avoid overfitting, we employ skip connections to enhance the performance of the proposed network. The number of feature maps in the first residual block is 16, which is half of that in the first layer within the decoder module. Similar to the encoder module, feature maps in the decoder module are accompanied by two residual blocks and upsampled twice using two 5×5 deconvolution kernel and a stride of 2. After each upsampling, feature maps of the same dimension within the encoder module are integrated and propagated backward using the skip-connection mechanism. Finally, the final high-resolution disparity map can be obtained using a separate upsampling layer.

2) Loss function: The loss function of the StereoVAE consists of the following two parts: 1) the difference between the high-resolution disparity map and the ground truth; 2) the difference between the output of the encoder module and its standard normal distribution. Here, since we normalize the image data in the first step, the value of the two loss functions must be expanded by a factor of 256 to ensure that the results

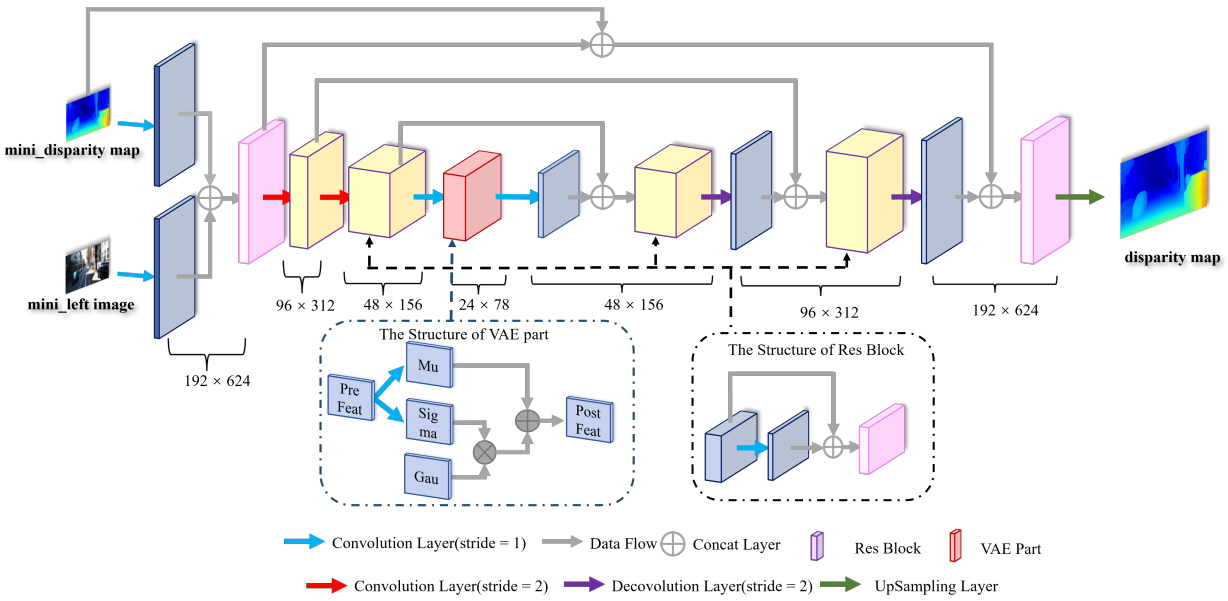

Fig. 2. StereoVAE Structure

TABLE I
STEREOVAE STRUCTURE

FEATURE EXTRACTION						
Layer	Input	Input Size	Output	Output size	Output Kernel Number	Kernel Size
1	X_0	H/2 * W/2	FE_1	H/2 * W/2	32	5 * 5
1	X_1	H/2 * W/2	FE_2	H/2 * W/2	32	5 * 5
ENCODER						
Layer	Input	Input Size	Output	Output size	output kernel number	Kernel Size
1	FE_1 & FE_2	H/2 * W/2	EN_1	H/4 * W/4	16	5 * 5
1	EN_1	H/4 * W/4	EN_1_R	H/4 * W/4	8	5 * 5
2	EN_1 & EN_1_R	H/4 * W/4	EN_2	H/8 * W/8	32	5 * 5
2	EN_2	H/8 * W/8	EN_2_R	H/8 * W/8	16	5 * 5
2	EN_2 & EN_2_R	H/8 * W/8	mu & sigma	H/8 * W/8	2	1 * 1
DECODER						
Layer	Input	Input Size	Output	Output size	output kernel number	Kernel Size
1	sigma * gaussian + mu	H/8 * W/8	D_1	H/8 * W/8	32	1 * 1
1	D_1 & EN_2 & EN_2_R	H/8 * W/8	D_1_R	H/8 * W/8	16	5 * 5
2	D_1 & EN_2 & EN_2_R & D_1_R	H/8 * W/8	D_2	H/4 * W/4	16	5 * 5
2	D_2 & EN_1 & EN_1_R	H/4 * W/4	D_2_R	H/4 * W/4	8	5 * 5
3	D_2 & EN_1 & EN_1_R & D_2_R	H/4 * W/4	D	H/2 * W/2	16	5 * 5
UP-SAMPLING						
Layer	Input	Input Size	Output	Output size	output kernel number	Kernel Size
1	D & X_0 & FE_1 & FE_2	H/2 * W/2	Y	H * W	1	5 * 5

are the same as those calculated in an 8-bit format. Then, the final loss function can be defined as follows:

$$Loss_{total} = 256 * (Loss_1 + Loss_2). \tag{9}$$

For $Loss_1$, we use the absolute error between the output and the ground truth and set the weight of the non-occluded points to twice that of the occluded points. Then, the specific loss $Loss_1$ can be defined as follows:

$$\begin{aligned} Loss_{noc} &= |gt_{noc} - Y|, \\ Loss_{occ} &= |gt_{occ} - Y|, \\ Loss_1 &= Loss_{noc} + Loss_{occ}. \end{aligned} \tag{10}$$

Here, Y is the output of the StereoVAE network, gt_{noc} is the ground truth without considering occlusion points, and gt_{occ} represents is the ground truth when considering occlusion points. Furthermore, $Loss_2$ can be expressed as follows:

$$\begin{aligned} Loss_2 = 0.5 * \{&(mu - 1)^2 - 1 \\ &+ exp^{(sigma-1)^2} \\ &- log(exp^{(sigma-1)^2})\} \end{aligned} \tag{11}$$

where *mu* and *sigma* represent the outputs of the encoder module. As the network iterates, both *mu* and *sigma* are expected to converge to 1.

IV. Experiments

In this section, we initially introduce the dataset and training conditions employed in our experiment, and then we

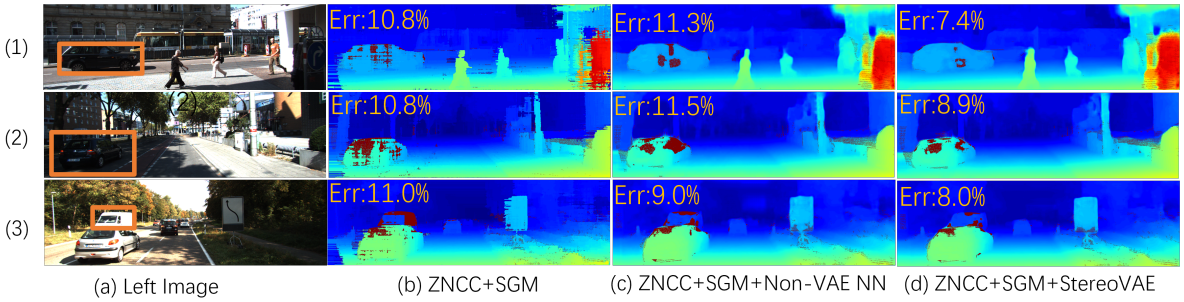

Fig. 3. Accuracy Comparison of StereoVAE using Different Methods. (1) Textureless region. (2) Specular region. (3) Brightness region.

present the experimental results to demonstrate the efficiency of the proposed system regarding the following aspects:

- comparison with traditional and non-VAE learning-based methods in terms of accuracy and processing speed;
- robustness testing of our StereoVAE on both GPUs (Jetson Tx2 and Jetson AGX Xavier) by refining the disparity maps generated using different traditional methods;
- performance comparison with existing embedded systems.

A. Preparation

The initial quarter size of disparity images was obtained by combining ZNCC and SGM with a half disparity range of 64. These two traditional methods were fully optimized using the CUDA platform to generate a coarse disparity map. The StereoVAE network was trained using the KITTI 2015 dataset, which is commonly used in autonomous driving research. The KITTI 2015 dataset contains 200 training image pairs and 200 testing image pairs. 160 of them were used for training and 40 for validation, respectively. The Adam optimizer was selected as the network optimizer, where β_1 was set to 0.99 and β_2 was set to 0.999. The initial learning rate was 0.0005. 1000 epochs were trained, and the learning rate was reduced by a factor of 3% every 10 epochs. The batch size was set to 1. Our StereoVAE has no constraints other than the loss function and an image cropping step.

B. Performance Evaluation

Traditional and non-VAE learning-based methods were compared with our method in terms of processing speed and accuracy. Table II shows the running speed of these three methods on both GPUs (Jetson TX2 and Jetson AGX Xavier). It is observed that the traditional method (ZNCC+SGM) maintains its advantage in terms of processing speed (28 fps and 79 fps), which is roughly twice faster than that of our StereoVAE (12 fps and 30 fps). However, our system can achieve real-time processing on the Jetson AGX Xavier GPU. The non-VAE learning-based method is slightly faster (14fps and 36 fps) than our StereoVAE, which means that the VAE structure does not spend too many computational resources. Figure 3 shows the accuracy

TABLE II
STEREOVAE EVALUATION

Method	Running time (Tx2)	Running time (AGX)
ZNCC+SGM	35.71 ms	12.64 ms
Non-VAE NN	72.1 ms	27.9 ms
StereoVAE	84.18 ms	29.88 ms

performance of these three methods under different image pairs. Regarding the error rate, ZNCC + SGM + StereoVAE $>$ ZNCC + SGM + non-VAE NN $>$ ZNCC + SGM fully demonstrates that our method is influential in the optimization. Since the traditional methods selected are based on pixel matching, the disparity maps they generated lack detailed optimization and contain much noise. In this aspect, learning-based methods are superior. Especially in the textureless, specular, and brightness regions (shown in orange boxes), traditional methods cannot produce satisfactory results due to the lack of matching features. In contrast, our VAE method exhibits a broader latent space expression and prediction ability.

C. Robustness

To evaluate the robustness of our StereoVAE, we compared its optimization performance using four combinations of traditional methods: S1(Census+DT), S2(Census+SGM), S3(ZNCC+DT), and S4(ZNCC+SGM). Figures 4 and 5 compare the accuracy of different methods that employ the KITTI 2015 validation and testing datasets, respectively. $Mini + Linear$ refers to directly using traditional methods to perform the stereo matching on small image pairs with a linear scaling. $Original$ represents the results obtained by applying direct matching on large image pairs. $Mini + StereoVAE$ represents the results obtained by applying the proposed method. Our StereoVAE improves the matching accuracy in all cases. The largest error rate reduction achieved by StereoVAE was 5.82%, which clearly demonstrates the high robustness level of the proposed method.

D. Comparison With Other Systems

Table III shows the comparison results of our system with other existing embedded stereo-vision systems. All systems were implemented on embedded GPUs, and achieved good performance. StereoDNN [23] exhibits the highest matching accuracy with only a 2.5% error rate. However, its processing

TABLE III
COMPARISON AMONG VARIOUS EMBEDDED STEREO-MATCHING SYSTEMS THAT EMPLOY THE KITTI 2015 DATASET

Error rate (%)	D1-bg		D1-fg		D1-all		Speed	
	All/All	Noc/All	All/All	Noc/All	All/All	Noc/All	GPU	fps
StereoNet[20]	4.3	—	7.45	—	4.83	—	Tx2	1
MADNet[21]	3.75	3.45	9.20	8.41	4.66	4.27	Tx2	4
RTS2Net[7]	3.09	—	5.91	—	3.56	—	Tx2	6
RTSMNet[8]	3.44	3.21	6.08	5.39	3.88	3.57	AGX	11
GPUBNN[9]	—	3.5	—	—	—	4.57	AGX	10
Res2tAC[3]	6.27	5.14	16.07	14.29	7.9	6.65	AGX	15
DWARF[22]	3.2	2.95	**3.94**	**3.66**	3.33	3.07	Tx2	1
AnyNet[4]	6.32	6.01	13.93	13.11	7.59	7.18	AGX	26
StereoDNN[23]	**2.7**	**2.1**	6.0	4.5	**3.2**	**2.5**	Tx2	1
Our StereoVAE	4.71	4.38	7.88	6.48	5.24	4.73	AGX	**30**

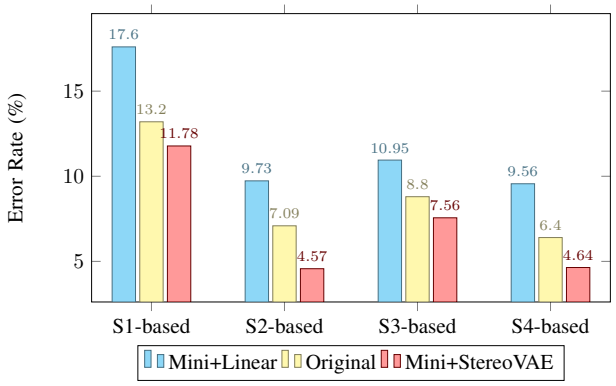

Fig. 4. Accuracy comparison among various systems that employ the KITTI 2015 validation dataset. Lower values mean better results.

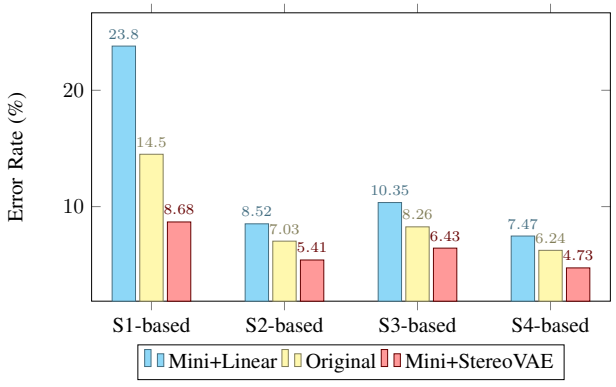

Fig. 5. Accuracy comparison among various systems that employ the KITTI 2015 testing dataset. Lower values mean better results.

speed is lower than 1 fps and thus it cannot meet the real-time requirements of embedded applications. On the other hand, our system exhibits the best processing speed in the embedded stereo-vision system list, while the accuracy is higher than that of the fast systems AnyNet [4] and Res2tAC [3], demonstrating that our system achieves a good balance regarding the embedded stereo-vision system performance.

V. CONCLUSIONS

In this study, we proposed a high-performance stereo-matching system implemented on a Jetson AGX Xavier GPU. The proposed hybrid structure includes of 1) the generation of a coarse disparity map using traditional methods and 2) a VAE-based neural network to upscale and refine the disparity map. Extensive experiments on the KITTI 2015 dataset, our system exhibits high processing speed and accuracy performance, indicating that our system combines the advantages of high processing speed of traditional methods and high accuracy of learning-based methods. However, there is still a gap in terms of accuracy compared with the current state-of-the-art methods, mainly because the proposed lightweight network lacks the required convolution kernels.

We plan to further compress our StereoVAE network through quantization to achieve increased processing speed and low memory usage, which can facilitate its application to other hardware platforms.

ACKNOWLEDGMENT

This work was partly supported by JSPS KAKENHI, Grant No. 21K17868, and JST CREST, Grant No. JPMJCR22M2.

REFERENCES

[1] Zbontar, Jure, and Yann LeCun. "Stereo matching by training a convolutional neural network to compare image patches." J. Mach. Learn. Res. 17.1 (2016): 2287-2318.
[2] Chang, Qiong, et al. "Efficient stereo matching on embedded GPUs with zero-means cross correlation." Journal of Systems Architecture 123 (2022): 102366.
[3] Ruf, Boitumelo, et al. "ReS2tAC¡½UAV-borne real-time SGM stereo optimized for embedded ARM and CUDA devices." Sensors 21.11 (2021): 3938.
[4] Wang, Yan, et al. "Anytime stereo image depth estimation on mobile devices." 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019.
[5] Gan, Wanshui, et al. "Light-weight network for real-time adaptive stereo depth estimation." Neurocomputing 441 (2021): 118-127.
[6] Chang, Jia-Ren, Pei-Chun Chang, and Yong-Sheng Chen. "Attention-aware feature aggregation for real-time stereo matching on edge devices." Proceedings of the Asian Conference on Computer Vision. 2020.
[7] Dovesi, Pier Luigi, et al. "Real-time semantic stereo matching." 2020 IEEE international conference on robotics and automation (ICRA). IEEE, 2020.
[8] Xie, Yun, Shaowu Zheng, and Weihua Li. "Feature-guided spatial attention upsampling for real-time stereo matching network." IEEE MultiMedia 28.1 (2020): 38-47.
[9] Chen, Gang, et al. "GPU-accelerated real-time stereo estimation with binary neural network." IEEE Transactions on Parallel and Distributed Systems 31.12 (2020): 2896-2907.

[10] Hernandez-Juarez, Daniel, et al. "Embedded real-time stereo estimation via semi-global matching on the GPU." Procedia Computer Science 80 (2016): 143-153.

[11] Menze, Moritz, and Andreas Geiger. "Object scene flow for autonomous vehicles." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015..

[12] Duggal, Shivam, et al. "Deeppruner: Learning efficient stereo matching via differentiable patchmatch." Proceedings of the IEEE/CVF international conference on computer vision. 2019..

[13] Bleyer, Michael, Christoph Rhemann, and Carsten Rother. "Patchmatch stereo-stereo matching with slanted support windows." Bmvc. Vol. 11. 2011.

[14] Zhang, Feihu, et al. "Ga-net: Guided aggregation net for end-to-end stereo matching." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[15] Riegler, Gernot, Matthias Rüther, and Horst Bischof. "Atgv-net: Accurate depth super-resolution." European conference on computer vision. Springer, Cham, 2016.

[16] Guo, Chenggang, Dongyi Chen, and Zhiqi Huang. "Learning efficient stereo matching network with depth discontinuity aware super-resolution." IEEE Access 7 (2019): 159712-159723.

[17] Wen, Yang, et al. "Deep color guided coarse-to-fine convolutional network cascade for depth image super-resolution." IEEE Transactions on Image Processing 28.2 (2018): 994-1006.

[18] Hirschmuller, Heiko. "Stereo vision in structured environments by consistent semi-global matching." 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). Vol. 2. IEEE, 2006.

[19] Di Stefano, Luigi, Massimiliano Marchionni, and Stefano Mattoccia. "A fast area-based stereo matching algorithm." Image and vision computing 22.12 (2004): 983-1005.

[20] Khamis, Sameh, et al. "Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[21] Tonioni, Alessio, et al. "Real-time self-adaptive deep stereo." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[22] Aleotti, Filippo, et al. "Learning end-to-end scene flow by distilling single tasks knowledge." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. No. 07. 2020.

[23] Smolyanskiy, Nikolai, Alexey Kamenev, and Stan Birchfield. "On the importance of stereo for accurate depth estimation: An efficient semi-supervised deep neural network approach." Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2018.