Correlate-and-Excite: Real-Time Stereo Matching via Guided Cost Volume Excitation

Antyanta Bangunharcana¹, Jae Won Cho², Seokju Lee², In So Kweon², Kyung-Soo Kim¹, Soohyun Kim¹

Abstract— Volumetric deep learning approach towards stereo matching aggregates a cost volume computed from input left and right images using 3D convolutions. Recent works showed that utilization of extracted image features and a spatially varying cost volume aggregation complements 3D convolutions. However, existing methods with spatially varying operations are complex, cost considerable computation time, and cause memory consumption to increase. In this work, we construct Guided Cost volume Excitation (GCE) and show that simple channel excitation of cost volume guided by image can improve performance considerably. Moreover, we propose a novel method of using top-k selection prior to soft-argmin disparity regression for computing the final disparity estimate. Combining our novel contributions, we present an end-to-end network that we call Correlate-and-Excite (CoEx). Extensive experiments of our model on the SceneFlow, KITTI 2012, and KITTI 2015 datasets demonstrate the effectiveness and efficiency of our model and show that our model outperforms other speed-based algorithms while also being competitive to other state-of-the-art algorithms. Codes will be made available at [https://github.com/antabangun/coex](https://github.com/antabangun/coex).

I. INTRODUCTION

Stereo matching aims to estimate depth from a pair of images [1], [2] and is an essential task in the field of robotics, autonomous driving, and computer vision. This task has various challenging issues such as occlusions, textureless areas, areas with repeating textures, thin or small objects, etc. With the advancements of deep learning algorithms, the accuracy of stereo matching algorithms has improved significantly; however, many accurate state-of-the-art models do not have fast processing speed for real-time applications [3]–[7]. Algorithms that focus on fast computations exist but often sacrifice accuracy to gain this advantage which may be the main reason why stereo cameras are not utilized more frequently in applications [8], [9] such as autonomous driving where fast computation is essential. If the efficiency of stereo matching algorithms can be improved from the current standard, stereo camera based depth perception can be an alternative to the expensive LiDAR sensors that are currently used in many self-driving algorithms [10].

Recent series of learning-based stereo matching algorithms [5], [11], [12] use left and right input images to construct a cost volume by computing the cross-correlation or concatenation of the features between from the two images. The correlation based approach reduces the input images’ feature vectors into cosine similarity values, giving a model with lower memory usage and faster runtime. However, this reduces the representation power of the neural network and often results in poor performance compared to the concatenation based cost volume.

In a volumetric approach, the computed cost volume is aggregated using 3D convolutional layers [13]. However, deep stacks of 3D convolutions are computationally expensive and memory inefficient [14]. Recent works have tried to improve the efficiency of the cost aggregation step using spatially varying aggregation [3], [5], [15]. While these works show improvements in accuracy, there is a significant increase in computational cost and memory consumption as well as additional complexity in the implementation of the proposed approaches.

We propose an efficient and straightforward way of improving cost aggregation by utilizing extracted image features using attention based approaches that have been shown to improve image classification networks [16], [17]. Given a cost volume feature map, Guided Cost volume Excitation (GCE) excites the cost volume channels with weights computed from the the reference image features. The computed weights are shared across the disparity channel, so the operation is lightweight and easy to implement. This module lets the 3D neural network layers to extract geometric features from the cost volume and the image-guided weights to excite the relevant features. We empirically show that this operation improves performance significantly without any significant additional computational cost. We show that this module allows correlation based cost volume to utilize image information and performs at a similar accuracy with the concatenation based model, allowing us to construct a

Fig. 1: D1-all% error on KITTI stereo 2015 leaderboard vs. frame rate. Our proposed method CoEx, shown in the red star, achieve competitive performance compared to other state-of-the-art models while also being real-time.
fast and accurate correlation based stereo matching model.

In volumetric based stereo matching models, soft-argmin is
the standard approach to compute the final disparity estimates,
and few works have been done to improve the soft-argmin
regression. The soft-argmin function computes the expected
value from a disparity distribution at each pixel obtained from
the cost volume aggregation. However, in many cases, the
disparity distribution can have multiple peaks e.g., on the
disparity map. We show that this simple yet novel idea
gives more accurate depth estimates and can be applied to
any volumetric based model.

With our proposed ideas, we construct an end-to-end real-
time stereo matching network that we call CoEx (Correlate-
and-Excite). We sum up our contributions and list them as
follows:

1) We present Guided Cost volume Excitation (GCE) to
utilize extracted feature map from image as guidance
for cost aggregation to improve performance.
2) We propose a new method of disparity regression in
place of soft-argmax(argin) to compute disparity from
the top-k matching cost values and show that it reliably
improves performance.
3) Through these methods, we build a real-time stereo
matching network CoEx, that outperforms other speed-
oriented methods and shows its competitiveness when
compared to state-of-the-art models.

II. RELATED WORKS

Recent works have focused on using deep Convolutional
Neural Networks (CNN) to improve stereo matching per-
formance. In [18]–[20], CNNs are used to obtain feature
representation for left and right images to be used for feature
matching, but cost aggregation is still done using traditional
means. DispNet [12] extended the idea to train an end-to-
end deep model to predict depth from stereo images by
introducing a correlation layer to construct the cost volume.
Following this, many more end-to-end works have been
proposed which can mostly be divided into either direct
regression or volumetric approach [21]. Direct regression
based methods use 2D convolutions on the cost volume
to directly compute the disparity map [22]–[24]. On the
other hand, volumetric based methods use 3D convolutions
to aggregate the cost volume by taking into account the
gemetric constraints [11], [13], [14], [21] and stacking 3D
convolutions in an hourglass architecture.

Recently, more works have focused on improving the
efficiency of 3D convolutions in the aggregation step. Two
notable works GANet [5] and CSPN [3] use spatially depen-
dent filters to aggregate cost. These methods have achieved
a higher accuracy using spatially dependent 3D aggregation
but at the cost of a higher computation time. Inspired by
the strengths and drawbacks of these approaches, we base
our model on spatially dependent 3D operation but focus
on speed and efficiency. On the other hand, StereoNet [25]
focused on building a real-time stereo matching model, and
like many others, do so by sacrificing its accuracy. Recently,
the accuracy of works [26], [27] on real-time stereo matching
models are getting closer to the best performing models.

The volumetric based approaches mentioned above outputs
a distribution of matching cost values at each disparity
level for every pixel. The final disparity estimates are then
computed by taking the expected value of the distribution
using a soft-argmin operation. As a result, the network is
only indirectly trained to produce a disparity distribution
and can fail in ambiguous regions. There have been few
works improving the soft-argmin disparity regression. Recent
studies AcfNet [28] and CDN [29] train the network to
produce better unimodal distribution by introducing novel
loss functions. This work presents a new method that builds
upon the soft-argmin operation itself and improves the overall
disparity regression.

III. METHOD

A deep learning based end-to-end stereo matching network
consists of matching cost computation, cost aggregation, and
disparity regression. We present a novel GCE and top-k soft-
argmin disparity regression module that can be integrated into
volumetric based baseline stereo approaches, both without
adding significant computation overhead to the baseline stereo
matching model. A real-time end-to-end stereo model is built
using the proposed modules, shown in Fig. [2] that achieves
competitive performance to the state-of-the-art. We describe
each of the components in detail in the following subsections.

A. Matching cost computation

Given a left and right input stereo image pair $3 \times H \times W$
feature maps are extracted from both of them using a shared
feature extraction module. We use MobileNetV2 [30] as our
backbone feature extractor for its lightweight property and
build a U-Net [31] style upsampling module with long skip
connections at each scale level. From this feature extraction
module, features at each scale are extracted for use later as
a guiding signal for spatially varying cost aggregations. To
construct the cost volume, feature maps extracted at the 1/4
scale of the left and right image are used with correlation
layer [12] to output a $D/4 \times H/4 \times W/4$ cost volume, where
$D = 192$ is the maximum disparity set for our network.

B. Guided Cost volume Excitation (GCE)

3D convolutions are used in modern architectures to
aggregate the constructed cost volume to allow the neural
network to capture geometric representation from the data.
Recent works [5], [32] have used spatially varying modules to
complement 3D convolutions and lead to better performance.
Specifically, weights are computed from the reference image
feature map to aggregate the 3D feature representation
computed from the cost volume. The modules compute
weights at each location for each pixel of interest and its
surrounding neighbors to allow for neighborhood aggregation
in a spatially dependent manner.

We argue that the 3D convolutions in a volumetric cost
aggregation already capture neighborhood information. A spa-
tially varying update of the cost volume feature map without
neighborhood aggregation is sufficient and is significantly
more efficient. To formulate it, for a cost volume with $c$
feature channels, we pass an image feature map at the same
scale into a guidance sub-network to output $c$ weights for
each pixel. With this formulation, the 3D convolutions capture
geometric information from the cost volume, and the guidance
weights excite the relevant geometric features. At scale $(s)$
of the cost volume:

$$
\alpha = \alpha(F^{2D}(I^{(s)}))
$$

$$
C_o^{(s)} = \alpha \times C_i^{(s)},
$$

where $F^{2D}$ is implemented using 2D point-wise convolution,
with $\sigma$ being the sigmoid function. The guidance weights are
shared across the disparity dimension, and the multiplication
in (1) is a broadcasted multiplication. This flow is shown
on the bottom left of Fig. 2. Since this module involves
excitation of cost volume features using weights computed
from the reference image feature map as guidance, we call
this module Guided Cost volume Excitation (GCE). This
module is extremely simple and straightforward, with only
a few operations added to the overall network; however, we
show in Sec. [IV-D] that adding GCE module can improve
the accuracy of our model significantly. In our CoEx model,
the cost aggregation architecture follows GC-Net [13], with
an hourglass architecture of 3D convolutions but with a
reduced number of channels and network depth to reduce
computational cost. The proposed GCE module is then added
at every scale of the cost volume (Fig. 2). The overall cost
aggregation module with GCE is detailed in Table [VI].
The module outputs a 4D cost volume at $1/4$ of the original image
resolution.

C. Top-k disparity regression

The 4D cost volume produced in the previous steps gives us
matching confidence values for each disparity level for every
pixel, which can be transformed into a probability distribution
by taking a Softmax across the disparity values. In previous
works, the soft-argmax operation is used to compute disparity
by taking the expected value over this distribution [13]:

$$
\hat{d} = \sum_{d=0}^{D} d \times \text{Softmax}(c_d)
$$

where $d$ is a predetermined set of disparity indices.

A disparity distribution where there is only a single peak
may give an adequate estimate for disparity predictions.
However, in some instances, there can be multiple peaks
or even a relatively uniform distribution. In these cases, the
expected value of the matching cost distribution can diverge
significantly from the actual ground truth value.

To alleviate this issue, instead of taking the expected
value of the whole distribution, we use only the top-k
values of the aggregated cost volume at every pixel. We call
this regression strategy top-k soft-argmax(argmin) disparity
regression. Specifically, at every pixel, we use the top-k
weights to compute the expected disparity value.

When $k$ equals the number of disparity of interest $D$, the
top-k regression is simply a soft-argmax operation [13]. When
$D > k > 1$, only the top-k values in each pixel are used
to compute the estimated disparity. This is done by masking
the top-k values and performing softmax on these values
to normalize them so that weights that sum up to 1 can
be obtained. These weights are then multiplied with their
corresponding disparity indices, while the remaining values
are masked out. The sum of the values are the weighted
average of the top-k disparity candidates. This operation can
be seen as similar to $k$-max pooling [33]. In the instance
where $k$ equals 1, the top-k regression becomes an argmax,
since the weight of the maximum index becomes a constant
at 1. When this is the case, the operation is not trainable, and
is why previous works resorted to using soft-argmax. Though
simple, we show through our experiments the effectiveness
of the top-k soft-argmax regression.

Using the top-k regression to compute the disparity map
at the full resolution requires a large amount of additional
computation time, as shown in Sec. [IV-D]. To mitigate this,
we design our model to compute the disparity regression at
1/4 of the input image resolution. Finally, the output disparity prediction is upsampled to the original input image resolution. Following the footsteps of [34], the final disparity estimate at each pixel in the upsampled resolution is obtained with a weighted average of a $3 \times 3$ “superpixel” surrounding it. Another CNN branch predicts the weights for each superpixel.

We train the network in a fully supervised end-to-end manner using $\text{smooth}_1$ loss function. Our final loss function is as follows:

$$L(d_{GT}, \hat{d}) = \frac{1}{N} \sum_{i=1}^{N} \text{smooth}_1(d_{GT, i} - \hat{d}_i),$$

given,

$$\text{smooth}_1(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases},$$

where, $N$ is the number of labeled pixels, $d_{GT}$ and $\hat{d}$ is the ground truth and predicted disparity respectively.

### IV. Experiments

In this section, we explain in detail the implementation details and training of our Correlate-and-Excite (CoEx) network, show through extensive experiments and ablations the effectiveness of our approach, and include detailed discussions on our method.

#### A. Datasets and Evaluation metrics

To test the effectiveness of our approach CoEx, we conduct experiments and evaluations on the following datasets: SceneFlow [12], KITTI Stereo 2012 [35], and KITTI Stereo 2015 [36].

SceneFlow is a synthetic dataset consisting of 35,454 training images and 4,370 testing images. The disparity range starts from 1 to 468, with all images having a size of $W = 960$, $H = 540$. We use the ‘finalpass’ version of the dataset. Only pixels with disparity values lower than our maximum disparity of 192 are used for training and evaluation. The end-point-error (EPE), which is the average difference between the predicted and ground truth, is used as a reporting metric.

KITT2012 and KITT2015 datasets are real-world datasets with sparse ground truth obtained from a LiDAR sensor. We divide the training data into 90% training and 10% validation set. KITT2012 uses ‘Out-All’, the percentage of erroneous pixels in total for an error threshold of 3 pixels, for its metric. For KITT2015, we show the ‘D1-all’ metric reported on the leaderboard, which is the percentage of all labeled pixels’ stereo disparity outliers.

#### B. Implementation details

We use the MobileNetV2 pre-trained on ImageNet [37] as listed in Sec. III-A for our feature extractor backbone. The use of ImageNet pre-trained model allows for faster convergence during training. We implement our model using PyTorch and use the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) as our optimizer with Stochastic Weight Averaging (SWA) [38]. We randomly crop images to size $W = 576$, $H = 288$ for training.

On the SceneFlow dataset, we train our model for 10 epochs with a learning rate $1 \times 10^{-3}$ for the first 7 epochs and $1 \times 10^{-4}$ for the remaining 3 epochs with a batch size of 8. For our experiments on the KITTI dataset, we use a model pre-trained on the SceneFlow dataset and finetune the model on the KITTI dataset for 800 epochs with an initial learning rate of $1 \times 10^{-3}$ decaying by a factor of 0.5 at epochs 30, 50, and 300. The Nvidia RTX 2080Ti GPU is used for training and testing.

#### C. Performance of CoEx

We show the comparisons of our model to the existing state-of-the-art in Table I. Note that KITT2015 results are all from the KITT Stereo Matching Leaderboard, and the SceneFlow EPE values, as well as the runtime, are the values reported in each work. Among the speed based models, StereoNet is the fastest performing model with a runtime of 15 ms. However, StereoNet’s accuracy on SceneFlow and KITT2015 is considerably less than CoEx, with differences being 0.411 EPE for SceneFlow and 2.7% on KITT2015.

As runtime comparisons in different hardware do not give a fair comparison, we compare the runtime breakdown of LEAstereo [21] and AANet [27] with our model tested on the same hardware (RTX 2080Ti) using the official open-source models in Table II The cost aggregation part includes cost volume construction and disparity regression. Our model is 3.3× faster than AANet while giving 0.18 EPE lower and 0.46% better KITT2012 3px out-all% and 0.42% better D1-all% on KITT2015. AANet+ added more focus towards disparity refinement to improve accuracy without sacrificing speed at the cost of a high number of network parameters at
Fig. 3: Qualitative results on KITTI 2015 test set. Error in orange corresponds erroneous prediction.

Fig. 4: Disparity distributions of models trained with different choice of $k$ in top-$k$ regression. Dashed red line is the estimated disparity and the solid green line is ground truth disparity.

### TABLE III: Ablation study of GCE and top-$k$ soft-argmin regression integrated into base models on SceneFlow ‘finalpass’ with the EPE metric (lower is better).

| Base model | Cost volume | GCE | Top-k reg | EPE (ms) |
|------------|-------------|-----|-----------|----------|
| PSMNet     | ✓           | ✓   | ✓         | 0.7437   |
|            | ✓           | ✓   | One       | 0.8176   |
|            | ✓           | ✓   | One       | 1.053    |
|            | ✓           | ✓   | One       | 0.8798   |
|            | ✓           | ✓   | One       | 0.8285   |
|            | ✓           | ✓   | One       | 0.8088   |
|            | ✓           | ✓   | One       | 0.7653   |
|            | ✓           | ✓   | One       | 1.108    |
| CoEx       | ✓           | ✓   | ✓         | 0.7552   |
|            | ✓           | ✓   | ✓         | 0.8262   |
|            | ✓           | ✓   | ✓         | 0.7928   |
|            | ✓           | ✓   | ✓         | 0.7942   |
|            | ✓           | ✓   | ✓         | 0.8242   |
|            | ✓           | ✓   | ✓         | 0.7426   |
|            | ✓           | ✓   | ✓         | 0.7782   |
|            | ✓           | ✓   | ✓         | 0.7185   |
|            | ✓           | ✓   | ✓         | 0.7115   |
|            | ✓           | ✓   | ✓         | 0.6854   |

8.4M compared to our 2.7M. Our model does not use any post aggregation refinement and still gives similar accuracy while being 3× faster than AANet+.

### D. Ablation study

We perform ablation studies on the SceneFlow dataset to study the influence of the proposed modules. We integrate GCE and top-$k$ soft-argmin regression into baseline stereo matching models. For this ablation study, we used the baseline PSMNet and CoEx model (Table III). Note that PSMNet uses a concatenation of the feature representations between the left and right images to construct cost volume. Concatenation allows the neural network to have a stronger representation power than correlation based cost volume construction that reduces the feature map to a single value of cosine similarity for each match. Replacing the concatenation in PSMNet to correlation reduces the accuracy as expected. However, adding only a single GCE layer into the correlation based PSMNet, indicated by ‘One’ in Table III brings the accuracy to a similar value to the concatenation based PSMNet, indicating that GCE enable the network to utilize image feature representations that is missed by correlation. In addition, the use of correlation also reduces the computation time significantly.

In PSMNet, the cost volume is upsampled to the original input image resolution and the maximum disparity value is
at $D = 192$. We test top-$k$ soft-argmin regression in PSMNet with $k$ between 2 to 192. We found that reducing $k$ from the original value of $k = 192$ generally improves performance up to a point. The accuracy degrades when $k$ is set too low, perhaps due to a lack of gradient flow in backpropagation. Moreover, performing sorting to obtain the top-$k$ values in the full cost volume resolution proves to be too computationally costly.

This motivated us to compute our disparity regression in the CoEx model at 1/4 the input image resolution and utilize the superpixel upsampling Sec. III-C to obtain the disparity map at the original resolution. Note that in CoEx, $k = 192/4 = 48$ is the maximum value of $k$. We show in Table III adding top-$k$ soft-argmin regression to CoEx hardly increases the computation time and gives better accuracy when lower $k$ values are used.

Table III also shows the performance gain when GCE is integrated at every scale level (Fig. 2), indicated by ‘Full’. Our best model is obtained when full GCE integration and top-2 soft-argmin regression are added into the base CoEx model. Notice that the two proposed modules only add 1ms of computation overhead from the base model but gives 0.17 lower test EPE.

1) GCE: We investigated two approaches to use the reference image as a guide for cost volume aggregation. The first is a simple addition between image features and cost volume features with a broadcasted operation, which effectively acts like a UNet style skip-connection. The second is based on excitation and is the proposed GCE module. The test comparison between the two on the SceneFlow dataset is shown in Table IV. Addition based skip-connection does give a slight accuracy improvement to the baseline. However, we found cost volume excitation a much more effective way of utilizing image features in cost aggregation.

We compare the GCE module that performs spatially varying local aggregation with a similar spatially varying operation that involves neighborhood aggregation. To do this, we formulate a neighborhood as a graph and use graph convolution to aggregate the nodes surrounding the center node of interest, where the graph edges are spatially varying and computed from the reference image feature map. The details of this graph-based aggregation are given in the Appendix. Table V shows that a simple excitation of the cost volume feature using a GCE module is performs better and more efficient than the implemented neighborhood spatially independent aggregation.

2) Top-2 disparity regression: To further illustrate how top-$k$ regression improves compared to soft-argmin regression, we plot the disparity distribution, produced from the output of cost aggregation, of models trained with each $k$ value. Fig. 3 illustrates 3 cases where a lower $k$ value in top-$k$ regression outperforms the baseline soft-argmin method. In the left most plot, the candidate disparities have a unimodal distribution. The middle case shows when there are 2 possible peaks, and the rightmost case shows the case when the distribution is relatively flat. In all these cases, the model trained using top-2 distribution is able to use only the peak matching values and is able to suppress values far away from the correct matching peak, resulting in a more accurate estimate.

Then how well would models trained with full soft-argmin perform when we replace this regression module with top-$k$ soft-argmin at test time? We provide experimental results for this test in Table III and found no improvement in the accuracy. The models need to learn to use the top-$k$ soft-argmin regression during training.

V. Conclusion

This paper introduces a new real-time stereo matching model that leverages spatially dependent cost aggregation that we call CoEx. We show that spatially varying aggregation can be performed in a lightweight and straightforward fashion to improve performance. We also show how a direct use of top-$k$ values can improve the soft-argmin disparity regression. We believe that the incredible speed of our method, where it is fast enough for real-time applications, can be a springboard for future real-time stereo matching research in real-world application settings.

APPENDIX

A. Detailed architecture

The detailed cost aggregation module is shown in Table VI. $s$ and $p$ are stride and padding sizes for the convolution kernels respectively. $I^{(5)}$ is the feature map of the left image obtained in the feature extraction stage at scale $(s)$.

| No. | Layer Setting | Input |
|-----|---------------|-------|
| 1   | correlation layer | $I^{(4)}$ (Left and Right) |
| 2-1 | conv3d 3 x 3 x 3, 8 | [1] |
| 2   | GCE | [2-1] and $I^{(4)}$ |
| 3-1 | conv3d 3 x 3 x 3, 16, $s = 2$ | [1] |
| 3   | GCE | [3-1] and $I^{(8)}$ |
| 4-1 | conv3d 3 x 3 x 3, 16, $s = 2$ | [1] |
| 4   | GCE | [4-1] and $I^{(16)}$ |
| 5-1 | conv3d 3 x 3 x 3, 48, $s = 2$ | [1] |
| 5   | GCE | [5-1] and $I^{(32)}$ |
| 6-1 | deconv3d 4 x 4 x 32, $s = 2$, $p = 1$ | [5] |
| 6-2 | conv3d 3 x 3 x 32 | [6-1] |
| 6   | GCE | [6-2] and $I^{(16)}$ |
| 7-1 | deconv3d 4 x 4 x 16, $s = 2$, $p = 1$ | [6] |
| 7-2 | conv3d 3 x 3 x 16 | [7-1] |
| 7   | GCE | [7-2] and $I^{(8)}$ |
| 8   | deconv3d 4 x 4 x 1, $s = 2$, $p = 1$ | [7] |

TABLE VI: Cost aggregation module.

B. Neighborhood aggregation

There are multiple previously proposed methods performing spatially varying aggregation that utilizes the neighborhood information [5], [15], [32]. To compare GCE with a module that computes spatially varying aggregation of the neighbors, we formulate a module that performs image-guided neighborhood aggregation. Given a voxel of interest at pixel location $i$ and its neighbors $j \in N(i)$ in a $1 \times n \times n$ window,
we compute the feature update of cost volume at \( i \) as follows:

\[
\begin{align*}
\tilde{m}_i^{(s+1)} &= \sum_{j \in \mathcal{N}(i)} e_{ji} \odot C_j^{(s+1)}, \\
C_j^{(s+1)} &= \xi \left( W_1 c_j^{(s)} + W_2 m_i^{(s+1)} + b \right),
\end{align*}
\]

(5)

where \( \odot \) represent element-wise product and \( \xi \) is an activation function. \( e_{ji} \) is the edge weight (or affinity in \cite{32}) of \( j \) to \( i \), and it is computed using MLP on the image features at \( i \) and \( j \), and also the encoding of the relative position \( p_i - p_j \) of the neighbors:

\[
\begin{align*}
\hat{e}_{ji} &= \text{MLP}(|I_i^{(s)}||I_j^{(s)}||\text{MLP}(p_i - p_j)|) \\
e_i &= \exp \hat{e}_{ji} / \sum_{j \in \mathcal{N}(i)} \exp \hat{e}_{ji},
\end{align*}
\]

(6)

where we use softmax (2nd line of the equation) to normalize the edge weights at each feature channel \( c \). In this work, Deep Graph Library (DGL) \cite{39} is used to implement the neighborhood aggregation as a graph.

For image feature map with \( c \) channels and cost volume of size \( c \times d \times h \times w \), GCE requires the following computation cost:

\[
(c_1 \times c \times h \times w) + (c \times d \times h \times w)
\]

(7)

Where the left part of the equation is the cost to obtain spatially varying weights, and the right part is the self-update. In contrast, if we write down the cost of weight computation and update of neighborhood aggregation in a \( 1 \times n \times n \) neighborhood, in the simplest form of weight computation where it computes weight by a point-wise convolution, it would require a computation cost of at least:

\[
(c_1 \times n \times n \times c \times h \times w) + (n \times n \times c \times d \times h \times w).
\]

(8)

Even in the simplest form, it would require \( n \times n \) more times than GCE.

REFERENCES

[1] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” IEEE Trans. on Patt. Anal. and Mach. Intel., 2007.

[2] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” International journal of computer vision, 2002.

[3] X. Cheng, P. Wang, and R. Yang, “Depth estimation via affinity learned with convolutional spatial propagation network,” in ECCV, 2018.

[4] Z. Yin, T. Darrell, and F. Yu, “Hierarchical discrete distribution decomposition for match density estimation,” in CVPR, 2019.

[5] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, “Ga-net: Guided aggregation net for end-to-end stereo matching,” in CVPR, 2019.

[6] S. Lee, S. Im, S. Lin, and I. S. Kweon, “Learning residual flow as dynamic motion from stereo videos,” in IROS, 2019.

[7] X. Gu, Z. Fan, S. Zhu, Z. Dai, F. Tan, and P. Tan, “Cascade cost volume for high-resolution multi-view stereo and stereo matching,” in CVPR, 2020.

[8] S. Lee, S. Im, S. Lin, and I. S. Kweon, “Learning monocular depth in dynamic scenes via instance-aware projection consistency,” in AAAI, 2021.

[9] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” in CVPR, 2017.

[10] S. Hwang, N. Kim, Y. Choi, S. Lee, and I. S. Kweon, “Fast multiple objects detection and tracking fusing color camera and 3d lidar for intelligent vehicles,” in URAI, 2016.