Soft Cross Entropy Loss and Bottleneck Tri-Cost Volume For Efficient Stereo Depth Prediction

Tyler Nuanes[†,‡] Matt Elsey[‡] Aswin Sankaranarayanan[†] John Shen[†]

[†]Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA
[‡]Light, 725 Shasta St, Redwood City, CA
Contact: tnuanes@andrew.cmu.edu

Abstract

Real-time, robust, and accurate stereo depth-prediction algorithms deliver cutting-edge performance in applications ranging from autonomous driving to augmented reality. Many state-of-the-art approaches produce subpixel error and subsecond runtimes on commodity hardware, but improving even these remains an area of active research. We focus on improving accuracy and efficiency in stereo-based depth prediction by contributing two generic techniques to improve performance and runtime. First, we propose encoding the ground truth disparity as a discrete distribution that can be trained via cross-entropy loss. Specifically, we use the minimum variance and unbiased 'Soft' encoding, where two adjacent bins are weighted so the expected value is ground truth. We demonstrate that training with cross entropy loss using this encoding decreases error rate by 10% *on synthetic and LIDAR datasets over the more popular regression losses such as Huber and MAE. Second, we propose a bottleneck tri-cost volume composed of the sum of absolute difference of the features as well as two reference channels. Replacing the standard 64-channel concatenation popular in state-of-the-art networks with this 3-channel cost-volume maintains metric performance and can reduce runtime by over* 22% *on PSM-Net architectures.*

1. Introduction

As an enabling technology, real-time, robust, and accurate depth prediction stands to improve occlusion interactions in augmented reality [1], lower costs for 3D object detection in autonomous vehicles [2], provide 3D mapping of scenes for virtual exploration [3], and advance widespread 3D visualization of objects [4]. As such, over the last few years, the computer vision community has witnessed an explosion of research in depth prediction via monocular, stereo, and multiview algorithms. We focus on stereo approaches, where cost volume architectures are pervasive in state-of-the-art [5–12].

Kitti 2015 Test Set	D1	Time (s)
PSM [5]	2.32	0.41
PSM-BTC [ours]	2.09	0.32

Table 1: State-of-the-Art Comparison. We show a 10% D1 improvement and 22% speed improvement with our proposed changes. BTC, or Bottleneck Tri-Cost, is a reference to our use of a bottleneck tri-cost volume and Soft cross entropy loss to learn the probability distribution directly.

The design of stereo networks commonly requires prediction of a probability distribution over a cost volume, a geometry-based structure from classical stereo vision. A cost volume is a 4D tensor of disparity, height, width, and costs (or features) where the disparity dimension typically accounts for methodical single-pixel shifts of the one of the stereo images against the other. After networks process the cost volume, a Softmax is taken over the disparity dimension to produce a discrete probability distribution. Many papers use the expected value, commonly termed the "Soft argmax" by Kendall *et al*. [13], over this distribution to estimate the true disparity of each pixel [5–11, 14–16]. Typically, regression losses, such as mean absolute error or Huber loss, train the network so the expected value has small deviation with ground truth, but the learned distribution may be multi-modal due to depth edges, repeating structure, or low-texture areas, resulting in a degraded expectation [10, 17–20]. Unfortunately, such regression losses suffer one-to-many relationship between the expected value and the probability distribution, which can add local minima noise to the gradient and degrade learning.

We argue that instead of learning expected value via regression, the probability distribution should be directly

learned through a categorical cross entropy (CCE) loss and a local expected value should predict the disparity. There has been some interest in pursuing this in the literature, but prior work suggests that regression loss performance is similar or superior [13, 18, 19]. However, we believe this result is strongly influence by the choice of the ground truth encoding. Thus, we propose a minimum variance, unbiased 'Soft' encoding whose expected value equals the ground truth disparity and demonstrate improved performance over regression losses with this technique.

In addition to proposing a new loss function, we also investigate efficient cost volume design. Existing literature uses a variety of architectures and multiple studies have demonstrated trade-offs between memory consumption, runtime, and metric quality. For instance, PSM Net concatenates 32 features from both views to allow the network to learn a distance metric [5]. AA Net uses a single cost, a dot product with no additional features [11], while GWC Net concatenates 12 features from each view and 40 correlation costs, each generated from 8 features, to improve performance [6]. Rao *et al*. use per-channel variance of 32 features in NLCA Net [8]. Several papers use single-channel cost-volumes to achieve real-time performance, but doing so often result in lower metrics, as shown in Table 2. We introduce a solution in the form of an efficient tri-channel cost volume, one we term the "bottleneck tri-cost volume" which maintains performance with standard 64-channel concatenation cost volumes at a fraction of the memory and computational cost.

On PSM-Net, we demonstrate a 10% decrease in the error rate from our proposed Soft encoding and a 22% faster runtime from the bottleneck tri-cost volume on the Kitti2015 benchmark (Table 1).

1.1. Contributions

We make the following contributions to benefit cost-volume stereo networks.

C1. Experiments showing a simple minimum variance, unbiased 'Soft' ground truth encoding for cross-entropy loss enables networks to learn more accurate probability distributions than standard regression losses. At inference, this translates to 10% lower error rate across synthetic and LIDAR-based datasets.

C2. Evidence that a bottleneck tri-cost volume, constructed with sum of absolute differences and 2 reference features, provides performance comparable to cost volumes 21 times larger in memory. During training, this enables larger patch or batch sizes. At inference, this translates to a 22% faster inference time on PSM-Net.

We report a series of experiments comparing loss functions and cost volume architectures to gain insight and guide development of the above contributions. These experiments evaluate ℓ_2 (mean squared error), ℓ_1 (mean absolute error), Huber (smooth ℓ_1), and categorical cross entropy (CCE) loss with Guassian, Laplacian, Hard, and Soft encoding of the ground truth. For the cost volume, we assess several distance metrics, including the absolute difference, euclidean distance, variance, and correlation functions. We consider these metrics per-channel and over all channels. We evaluate standard concatenation and compact representations. In order to demonstrate performance on state-of-the-art, we perform an ablation study on PSM-Net, comparing performance differences between the base network and ones trained with each of our proposed changes.

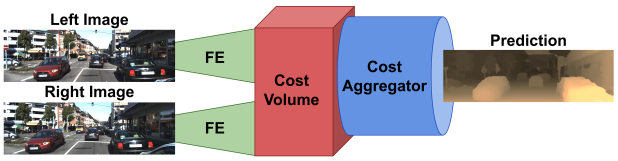

Figure 1: Example of a stereo network. FE stands for the feature extractor, which is Siamese, performing the same operations on the left and right images. The cost-volume is a 4D tensor generated without any learned parameters. The cost aggregator processes the 4D tensor to make a prediction. Image 110 from the Kitti 2015 test set.

2. Background

Many stereo papers use a 3-stage network composed of feature extraction, cost volume, and cost aggregation, as shown in Figure 1. Typically, the feature extractor is a Siamese network that uses shared weights to extract matched features from a pair of images. The cost volume generates a comparison between the reference and secondary features, over the range of disparities under consideration. Finally, the cost aggregator processes the cost volume to make a final dense prediction for disparity.

The majority of the top performing networks on the Kitti 2015 benchmark today use a cost-volume network, including NLCA Net [8] AM Net [9], ACF Net [10], GA Net [7], CSP Net [21], and optical expansion net [22]. Only SUW-Learn does not have a cost volume; instead it uses monocular time-series data [23], which includes an unsupervised warping loss between time-steps. While single-frame monocular approaches have made remarkable strides over the last few years, monocular predictions tend to report worse metrics than stereo predictions [24]. In fact, Ranjan *et al*. demonstrated that developing an optical flow architecture with cost volumes leads to more robust performance in the face of adversarial attacks than an encoder/decoder architecture [25]. This work suggests that while processing cost volumes may be expensive, such representation is more robust than other approaches.

Real-time networks often have similar architectures to state-of-the-art, although they typically sacrifice low error rates for speed. For instance, networks focussd on compute speed, such as HD^3-Net [26], RTS^2-Net [16], AnyNet [15], and Stereo-Net [14], MAD-Net [27], Disp-NetC [28], AA-Net [11], and HIT-Net [12] achieve impressive FPS on the Kitti test set, as shown in Table 2. The majority of these networks, inspired by optical flow techniques, achieve high speed by generating multiple cost volumes in the style of PWC-Net [29], where each successive cost volume is generated at a higher resolution and attempts to refine the residual error of the lower resolution prediction. We do not directly address such architectures in this work, but our Soft encoding and bottleneck tri-cost volume are applicable to these networks.

Of particular note, Yin *et al*.'s optical flow network, HD^3, introduces 'Vector to Density' [26], a ground truth encoding scheme mathematically equivalent to the Soft encoding, and train with Kullback-Leibler (KL) divergence, which is equivalent to minimizing CCE. Yin *et al*. demonstrate proficiency training mutliresolution cost volumes and flow networks using Soft KL. However, their contribution is distinct from ours. Yin *et al*. work to show their unique network, when trained by Soft KL loss, is competitive with state-of-the art and near real-time performance. In contrast, we demonstrate Soft CCE loss is superior to standard regression losses. We hope this generalized insight helps researchers push forward state-of-the-art on their networks.

2.1. Cost Volume Architectures

In the literature, distance metrics used by cost volumes vary substantially, as demonstrated in Table 2. For instance, NLCA Net uses 32 variance features and reports 2-4% metric improvements over concatenation [8]. AM Net introduces an extended cost volume, including concatenation, multiplication, and absolute difference of each feature. They do not clarify the number of channels, but they do provide a series of experiments showing a steady improvement by combining these metrics [9]. GA Net, GC Net, and RTS Net simply concatenate reference and secondary features to produce a 64-channel cost volume [7, 13, 20]. GWC Net carries out a series of experiments on cost-volume construction with correlation groups. They work with 320 channels from the feature extractor, and show performance improves as they increase the number of groups from 1 to 160. Their best cost volume uses 40 correlation groups concatenated to 12 channels from each view, for a total of 64 channels [6]. Stereo Net creates a cost-volume of C features by taking the difference of channels between the two candidates.

Many networks optimized for speed use only a single channel in their cost volumes. For instance, Any Net uses the sum of absolute differences (SAD) [15]. DispNetC uses correlation at a low resolution and uses features from the reference image to upsample the prediction [28]. MAD Net generates correlation cost-volumes at multiple resolutions; at each higher resolution, MAD uses the lower-resolution prediction to warp the secondary image into the reference view, generates a new correlation cost-volume, and refines the disparity prediction.

Similarly, AA Net generates cost volumes from the correlation of all features at each resolution, and then uses a series of multiscale deformable convolutions to refine re-

Net	Losses	C_{CV}	D1	Time (s)
Select State-of-the-Art on Kitti 2015				
PSM [5]	Huber	64	2.32	0.41
GWC [6]	Huber	64	2.11	0.32
ACF [10]	Huber Focal Confidence	64	1.89	0.48
AM [9]	Huber Segmentation	$4C$	1.84	0.9
NLCA [8]	ℓ_1 SSIM Warping	32	1.83	0.44
GA [7]	Huber	64	1.81	1.8
CSP [21]	ℓ_1	64	1.74	1.0
Select Real-Time Networks				
Any [15]	Huber	1	6.2	0.097
Stereo [14]	Huber	C	4.83	0.015
MAD [27]	Photometric	1	4.66	0.02
DispNetC [28]	ℓ_1	1	4.34	0.06
RTS^2 [16]	Huber Segmentation	C	3.56	0.02
AA [11]	Huber	1	2.03	0.06
HIT [12]	ℓ_1 Huber Slant Confidence	1	1.98	0.015
Select Studies on the Distribution				
ES [30]	Gaussian	1	4.54	1.0
RTS [20]	Focal	64	3.41	0.02
GC [13]	ℓ_1	64	2.87	0.9
PS [18]	Laplacian	8	2.58	0.5
PSM-CD [17]	Wasserstein	64	2.29	0.4
NS [19]	ℓ_1 Laplacian	64	2.27	0.6
PSM-BTC [ours]	Soft	3	2.09	0.32
HD^3 [26]	Soft	$>C$	2.02	0.14

Table 2: Stereo Networks. D1 is reported on the Kitti 2015 test benchmark. C_{CV} is cost volume channels. C is the number of channels from the feature extractor. HD^3 includes feature correlation, reference features, and an embedding vector. Note: Some networks, such as HIT and HD^3 predict disparity at multiple resolutions.

sults [11]. HIT Net uses the sum of absolute differences to quickly produce disparity initializations at multiple resolutions and proceeds to refine these hypotheses with tiles of the disparity, slant, estimated cost, and 16 reference view features (at highest resolution), achieving close to the state-of-the-art in real-time [12]. Single cost networks introduce some questions. How much information is really lost by using a single cost? Are they making up for significant degradations with their unique aggregators or are large cost volumes an extremely inefficient component of stereo networks? Is AA Net's use of correlation a wiser choice than HIT Net's ℓ_1 distance?

2.2. Categorical Cross Entropy (CCE) Losses

As shown in Table 2, the top-performing stereo networks use regression losses. However, we contend that properly formulated CCE losses are more competitive than regression losses because stereo networks perform regression by first predicting the underlying distribution, and then calculating the expected value over this distribution. Thus, the regression suffers from a one-to-many relationship between the expected value and predicted distribution, as shown at the bottom of Figure 2. That is, the expected value of many plausible distributions can result in a given ground truth. Zhang *et al*. recognize this problem and suggest a series of constraints to improve performance in ACF Net [10]. Among other techniques, Zhang *et al*. use a focal cross-entropy loss with Hard one-hot encoding to improve the learned distribution. Similar to Zhang's work, Garg *et al*. focus on learning the distribution with a Wasserstein loss. They learn a Hard one-hot distribution and an offset to the ground truth, and demonstrate that this technique improves results on benchmark models [17].

Various CCE losses have been proposed for stereo networks. Luo *et al*. used an approximate Gaussian distribution in their 2016 ES Net [30]. In 2017, Kendall *et al*. argued that between Luo *et al*.'s Gaussian distribution, a Hard one-hot encoding, and ℓ_1 loss, ℓ_1 regression outperforms CCE losses in the long term on their GC Net, even though the cross entropy loss initially learns faster. Notably, GC Net is trained up to 120,000 iterations, whereas we train for 170,000 iterations. GC Net had an EPE of 2.5 px for ℓ_1, over 5.0 px for Hard CCE, and 5.4 px for Gaussian CCE [13]. In 2018, Tulyakov *et al*. used a Laplacian CCE to improve the 3PE metric on their PS Net and use sub-pixel MAP estimation to further improve results [18]; however, their results on EPE were worse for Laplacian CCE than ℓ_1 loss. In 2019, Lee and Shin reported that a focal loss centered around a Hard one-hot encoding could learn effectively in their RTS Net, which runs in 0.02 sec. Instead of sub-pixel MAP, they proposed using the Top K lowest cost disparities to estimate the disparity [20]. In 2020, Chen *et al*. used a Laplacian CCE loss to slightly improve the learned distribution of PSM Net [19].

2.3. Application to SOTA

As demonstrated in Table 2 and our background discussion, almost all state-of-the-art (SOTA) networks use cost volumes. Our bottleneck tri-cost volume is a drop-in replacement for other cost volumes. Depending on the architecture, single-cost networks, such as HIT-Net [12] and AA-Net [11] have specialized backends and may not be designed to use the bottleneck tri-cost volume as-is. HIT-Net already makes use of reference features whereas AA-Net expects a single feature for processing.

Our loss function is also widely applicable to SOTA. For most networks, it can be implemented as a drop-in replacement for Huber or MAE loss. On multistage networks, such as PWC-Network (which is originally implemented for optical flow prediction), where cost-volumes are generated at multiple resolutions around a narrow disparity window using warped features, Soft CCE is still applicable. The ground truth encoding would simply shift to encompass the correct disparity window at each refinement stage.

3. Methodology

3.1. Soft Categorical Cross-Entropy Loss

Our proposed Soft CCE encoding is a neighboring 2-bin probability distribution whose expected value equals ground truth. For instance, if the cost volume has 8 disparity levels starting at 0 px disparity with shifts of 1 px per step, and the ground truth disparity for a pixel is 0.4 px, the CCE ground truth would be $[0.6, 0.4, 0, 0, 0, 0, 0, 0]$, as shown in Figure 2. Soft encoding is the "ideal" distribution that could be learned via regression as it is the minimum variance, unbiased distribution for a given expected value.

We compare Hard, Gaussian, and Laplacian encodings from literature against our proposed Soft encoding in Figure 2. Hard is a one-hot encoding, with disparity rounded to its nearest index. Gaussian encodes the ground truth by forming a normalized Gaussian distribution centered around the true disparity with variance $\sigma^2 = \frac{2}{\pi}$ based on ES Net [30]. We create a normalized Laplacian distribution centered around the ground truth disparity with $b = 2$ based on on PS Net [18], resulting in a much larger bandwidth than the Gaussian distribution. Due to bin boundaries, there is no guarantee that the expected value over Hard, Gaussian, or Laplacian distributions equals ground truth.

Regression losses do not necessarily learn minimum-variance distributions, as demonstrated by [10]. For instance, a distribution learned from regression losses could be $[0.8, 0, 0.2, 0, 0, 0, 0, 0]$ since $0.2 * 2$ px. $= 0.4$ px, as shown in Figure 2. Additionally, since many different distributions can equal a given expected value, the gradient may not be very clear, resulting in slower training, as noted

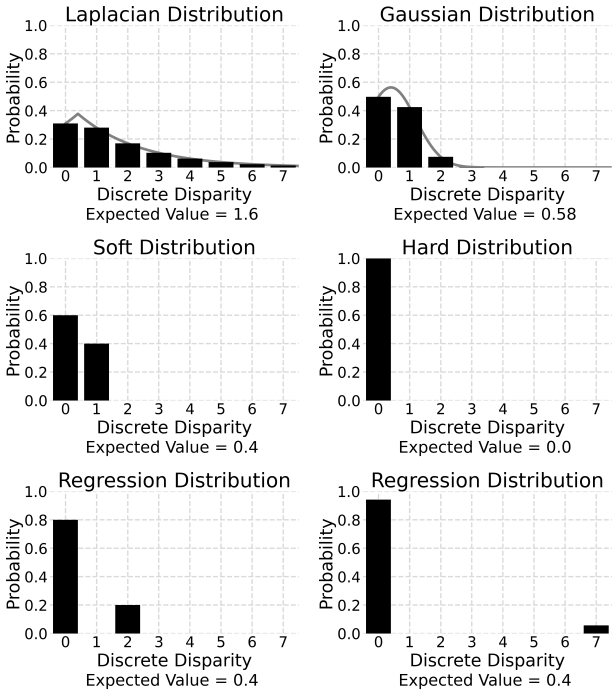

Figure 2: Minimum loss probability distributions for 0.4 px disparity. Bins are bounded by min (0 px) and max (7 px), which degrades Gaussian and Laplacian ground truth encodings. Gaussian and Laplacian distributions centered around 3.4 px would have small error in expected value. The two regression distributions both minimize the regression loss, demonstrating how minimization over regression has one-to-many solutions with multiple local minima.

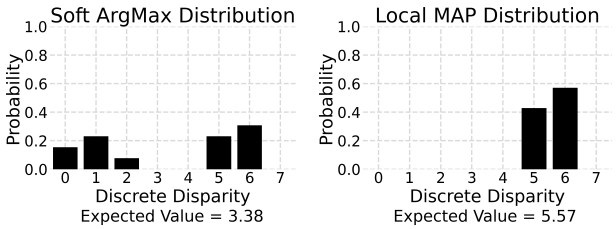

Figure 3: Predicted distribution for ground truth 5.6 pix disparity before and after applying local MAP re-normalization. The left probability distribution is the network's prediction with two local maxima probabilities, perhaps along a depth discontinuity. The expected value 3.4 pix deviates greatly from the true disparity, and will result in a blurred edge. The right probability distribution is the result after applying local MAP re-normalization with $\delta = 1$. As can be seen, the new expected value greatly reduces error with ground truth and results in a sharper edge.

by [13, 18].

The apparent drawback of Soft CCE is that, at training time, it penalizes all disjoint errors equally. For instance, a disjoint distribution producing 3 px error results in as much loss as disjoint distribution producing 100 px error, which intuitively seems undesirable, but actually makes Soft CCE more robust to challenging outliers.

For instance, suppose there are two pixels in an image, each with ground truth disparity Soft encoding $[0.3, 0.7, 0, 0, 0, 0, 0, 0]$ (0.7 pix). Let us assume the network makes the following two predictions: $\text{P1} = [0, 0, 0.6, 0.4, 0, 0, 0, 0]$ (2.4 pix) and $\text{P2} = [0, 0, 0, 0, 0, 0, 0.2, 0.8]$ (6.8 pix). Let's assume that for numeric stability, $\epsilon =$ 1e-7. The loss for each of these predictions is $\text{CCE Loss} = -(0.3 * log(\epsilon) + 0.7 * log(\epsilon)) = -log(\epsilon) = 16.12$. In comparison, $\ell_{1,\text{P1}} = |2.4 - 0.7| = 1.7$ and $\ell_{1,\text{P12}} = |6.8 - 0.7| = 6.1$. For MSE, $\ell_{2,\text{P1}} = 1.7^2 = 2.89$ and $\ell_{2,\text{P12}} = 6.1^2 = 37.2$. MAE loss puts more emphasis on correctly predicting the more challenging pixels than CCE Loss while MSE does this to an even greater extent. We compare training on MSE, MAE, and Soft CCE in Table 3. We show that MAE greatly outperforms MSE, suggesting that heavily weighing challenging outliers harms optimization.

In practice, we find that Soft CCE improves learning and propose three explanations:

1. Progress in stereo networks and datasets has reduced average prediction error to subpixel, enabling CCE to predict subpixel distributions accurately.
2. Soft CCE Loss is more robust than regression losses since it penalizes all disjoint predictions equally, spending fewer resources on challenging outliers.
3. The one-to-many relation between expectation and probability distribution results in noisy gradients with many local minima for regression losses, slowing training and limiting learning.

3.2. Local Maximum A Posteriori (MAP)

As detailed in Zhang *et al*. [10], Garg *et al*. [17], Tulyakov *et al*. [18], Chen *et al*. [19], and Lee *et al*. [20], the predicted disparity distributions may be multimodal, having various local minima due to repeating structure, depth boundaries, pixel noise, or low texture. In such situations, simple expected value may pull the prediction away from the true ground truth, as shown in Figure 3. Each prior work proposes their own solution to this problem, such as Top K [20] or Wasserstein distance loss [17]. The Top K approach would still be susceptible to multiple local minima, so we choose to use the subpixel MAP introduced by Tulyakov *et al*. to estimate disparity at inference [18] by the following equation:

$$d_{\delta}^{MAP} = \sum_{d=\hat{d}-\delta}^{\hat{d}+\delta} d \cdot \hat{P}(\mathbf{d} = d | \mathbf{x}^L, \mathbf{x}^R) \qquad (3.1)$$

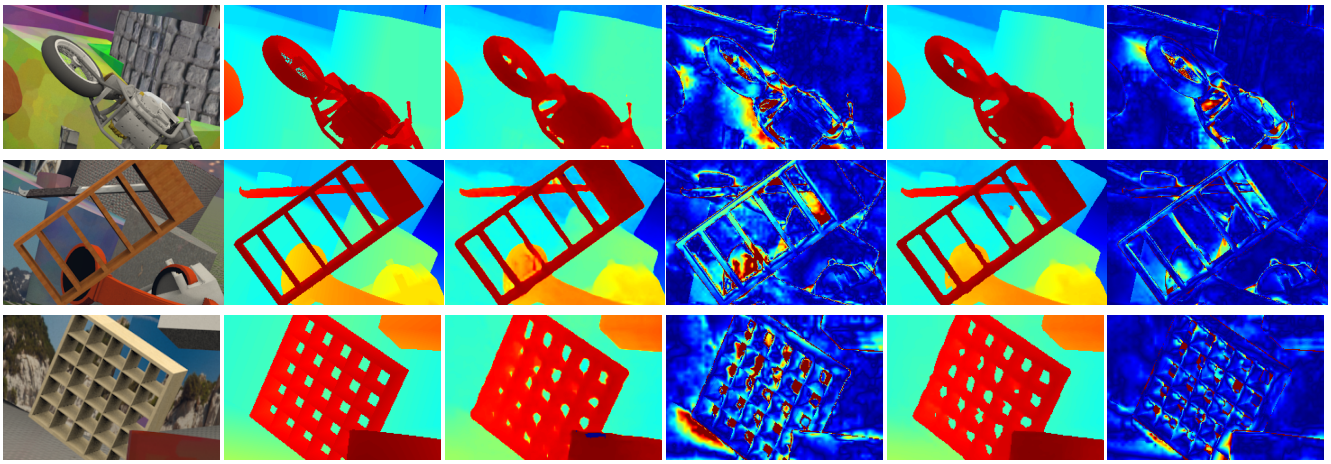

Figure 4: Examples of PSM-BTC [ours] inference on FlyingThings3DClean Test images A/0057/0009, B/0097/0009, & C/0062/0009. PSM-BTC uses local MAP while PSM uses regression in this example. These predictions are made at $\frac{1}{4}$ resolution and bilinearly upsampled before calculating the expected value. Note that local MAP ($\delta = 1$) with Soft CCE sharpens edges and better distinguishes depth discontinuities in complex scenes where multiple local cost minima may be present.

where d is a disparity step and the maximum probability disparity, $\hat{d}$, is

$$\hat{d} = \underset{0 \leq d \leq D}{\operatorname{argmax}} P(\mathbf{d} = d | \mathbf{x}^L, \mathbf{x}^R) \tag{3.2}$$

We do clarify that $\hat{P}$ is locally normalized from the full predicted distribution P by

$$\tilde{P} = \sum_{d=\hat{d}-\delta}^{\hat{d}+\delta} P(\mathbf{d} = d | \mathbf{x}^L, \mathbf{x}^R) \tag{3.3}$$

$$\hat{P}(\mathbf{d} = d | \mathbf{x}^L, \mathbf{x}^R) = \begin{cases} \frac{P(\mathbf{d}=d|\mathbf{x}^L,\mathbf{x}^R)}{\tilde{P}} & |\hat{d} - d| \leq \delta \\ 0 & \text{else} \end{cases} \tag{3.4}$$

When $\delta = \infty$, we sum over all predicted indices. We slightly modify Tulyakov's equation by specifying that when $\delta = \frac{1}{2}$, we only sum over two indices: the maximum probability index and the highest probability adjacent index. This additional capability enables intuitive comparison of our predicted probability distribution against the Soft-encoded ground truth. We only use subpixel MAP at inference time; training of regression losses occurs with $\delta = \infty$.

We demonstrate the benefits of local MAP trained with Soft CCE loss over a regression model trained with Huber loss in Figure 4. Near complicated depth boundaries, disparity edges tend to appear crisper under local MAP. To observe the improvements under local MAP, it is important to not interpolate over the depth dimension.

3.3. Bottleneck Tri-Cost (BTC) Volume

As covered in the background, cost-volume experiments have been performed before, but results are difficult to compare between disparate networks, training methods, and datasets. There is no consensus in the literature on how to construct the cost-volume efficiently. We aim to fill that gap and provide a more complete picture by performing a series of experiments where we examine different distance metrics, multi-cost volumes, and compact representations. The full list of experiments is in the appendix, and a truncated version highlighting the most relevant experiments appears here. Some of the cost volumes we consider are demonstrated visually in Figure 5.

Based on our experimental results, we propose the bottleneck tri-cost volume, which is composed of the sum of absolute differences (SAD) plus two separate reference features. This is 21 times smaller than the popular concatenation technique but has comparable performance. To generate this cost volume, we start with the standard 32 channel output for each view from the feature extractor. We increase the output channels to 34. For a given disparity, SAD is taken over 32 of the features. The final 2 features of the reference view are concatenated along the channel axis of the SAD result, while the final 2 features of the secondary view are discarded. The secondary view is then shifted to the next disparity step, and the process is repeated until the full 3-channel cost volume composed of both stereo and monocular cues is generated. This procedure does replicate the same 2 reference features across all disparity steps, resulting in an inefficient representation of the monocular infor-

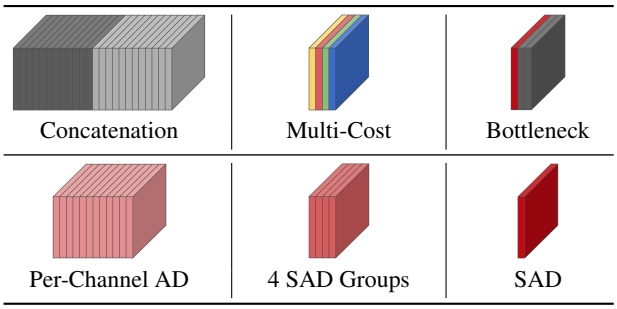

Figure 5: Cost Volume Visualization. In this example, the feature extractor outputs 12 channels for the reference and secondary views, represented by dark and light gray. Costs may be absolute difference (AD), correlation, variance, etc and are in color. SAD is sum of AD.

	MSE_1	Hu_5	MAE_3	L_1	Ha_1	G_3	S_5
EPE	2.72	1.37	1.29	1.62	1.46	1.21	**1.12**
D1	18.5	5.0	4.8	5.4	3.9	3.9	**3.8**
D½	46.6	17.2	14.3	30.9	31.4	20.6	**11.2**

Table 3: Select Lightweight SST Net inference results under local MAP at $\delta = 2$ on FlyingThings3D Test Set. Subscripts indicate the number of times each experiment was run & averaged. Minimum values per column are bold. Values within 5% of the minimum of each column are underlined. Hu is Huber regression loss, L is Laplacian CCE, Ha is Hard CCE, G is Gaussian CCE, and S is Soft CCE. In the appendix, you will find this table reproduced with results for $\delta = \{\infty, 3, 2, 1, \frac{1}{2}\}$.

mation and presenting opportunities for future research to improve performance.

3.4. D½ Error Metric.

Due to the improvement in EPE in recent years, we introduce the D½ metric, derived from the D1 metric. The D1 metric is an error rate that classifies all pixels over 5% relative error and 3 px absolute error as erroneous pixels. Instead of the typical 3 px threshold of the D1 metric, we use a 0.5 px absolute error threshold in the D½ metric, which is more discriminative of errors in distant regions where accuracy may be important for autonomous cars moving at highway speeds. D½ provides tighter 5% error bounds from 10 px to 60 px, where the D1 metric is dominated by the 3 px threshold.

4. Experiments

We perform a parameter sweep over loss functions and cost volumes on a lightweight network, known as Small Simple Training (SST) Net, which is effectively a smaller version of PSM-Net. For details of the network, setup, results, and additional experiments, please visit the appendix.

Experiments on SST Net support two main conclusions:

1. Soft CCE Loss outperforms regression losses as well as other ground truth encoding schemes for CCE Loss (Table 3). Soft CCE performance increases as local MAP δ decreases to 1 (see appendix).
2. SAD plus two reference features provides a baseline of performance that is difficult to meaningfully surmount (Table 4).

Based on these results, we chose to validate performance on a state-of-the-art network, PSM-Net, where we carry out a series of ablation studies comparing these two new techniques against the reference implementation. On these experiments, we follow the training methodology of PSM-Net [5].

4.1. Highlighted Experiments on SST Net

We evaluate various losses in Table 3, comparing results on FlyingThings3D when inferred under local MAP. For reference, $\delta = 2$ was chosen as it was near-minimum for each experiment. The appendix include additional experiments for those looking to observe how performance changes with δ. We find that for each metric we considered, Soft CCE outperforms every other considered loss.

We compare cost volume performance in Table 4. We demonstrate metric differences between single-cost and per-channel distance metrics. Single-cost networks appear to be limited by the lack of purely monocular cues. We particularly aim to highlight the increase in metrics as additional reference features are added to a single distance metric, up to 2, where metric gains level off. While generating a cost-volume composed of 96 channels is able to make modest gains, for many use cases, the bottleneck tri-cost volume has an excellent trade-off of metric quality with computational expense.

4.2. PSM-BTC Net Ablation Studies

For evaluation on PSM-BTC Net, we modify the original cost-volume of PSM Net and change trilinear upsampling to bilinear upsampling to enable Soft CCE training and local MAP to have meaningful results. We evaluate on the stacked hourglass & pyramid pooling model with half the dilation rate. We train according to PSM Net's specified methodology [5] and evaluate over the standard 192 pix for the ablation study and Kitti2015 benchmark; however, we only pretrain on the FlyingThings3DClean subset of the SceneFlow dataset. We performed the speed test in Table 1 on an NVIDIA Titan Xp, the same as PSM Net, for direct comparison.

Architecture	GFLOP	C_{CV}	EPE	D1	D½
Concat	636	64	1.30	5.3	15.8
AD	571	32	1.29	5.3	15.4
Muliply	571	32	1.46	5.8	16.7
Variance	571	32	1.36	6.0	16.3
Concat & Multiply	700	96	1.26	**5.0**	**14.4**
Concat & Variance	701	96	**1.25**	**5.0**	14.5
Concat & AD	700	96	**1.25**	5.1	14.7
SAD & Concat 4 RF	518	5	1.28	5.1	15.2
SAD & Concat 2 RF	513	3	1.27	5.1	15.3
SAD & Concat 1 RF	511	2	1.36	5.5	16.2
SAD	509	1	1.44	6.1	17.4
Euclidean	509	1	1.55	6.8	19.1
Correlation	509	1	1.55	6.1	17.0
Sum Variance	509	1	1.46	6.1	17.8

Table 4: Select SST Net inference results on FlyingThings3D Test Set. Concat is concatenation. RF are separate, additional reference features. C_{CV} is the cost volume channels. Minimum values are bold. Our bottleneck tri-cost volume is underlined. The appendix includes this table with additional experiments.

Loss	Cost Volume	EPE	D1	D½	>1 pix
Flying Things 3D Clean Test Set					
Huber	Concatenation	1.05	3.3	5.8	10.6
Huber	Bottleneck Tri-Cost	1.06	3.4	5.9	10.6
Soft	Concatenation	**0.94**	**2.7**	**3.9**	7.5
Soft	Bottleneck Tri-Cost	**0.94**	**2.7**	4.1	**7.4**
Kitti 2015 160 Image Training Split					
Huber	Concatenation	0.51	0.97	6.8	9.7
Huber	Bottleneck Tri-Cost	0.53	1.00	7.2	10.5
Soft	Concatenation	**0.45**	0.70	**4.8**	**6.9**
Soft	Bottleneck Tri-Cost	**0.45**	**0.64**	5.1	7.4
Kitti 2015 40 Image Validation Split					
Huber	Concatenation	0.72	1.97	10.4	16.4
Huber	Bottleneck Tri-Cost	0.71	2.02	10.7	17.1
Soft	Concatenation	0.69	1.90	**8.8**	**14.8**
Soft	Bottleneck Tri-Cost	**0.68**	**1.79**	9.2	15.1

Table 5: PSM-BTC Net ablation study under local MAP at $\delta = 1$ as D½ minimized here for all experiments. We use the same data splits reported by PSM-Net, which reported an EPE of 1.12 pix for FlyingThings3DClean and a D1 of 1.83% on the Kitti2015 validation split [5]. The best metric on each dataset is bolded.

Our ablation study demonstrates comparable metrics on the concatenation and bottleneck tri-cost volumes (Table 5).

Notably, the Kitti2015 dataset experiences a large degree of overfitting to the training set (where Soft CCE results are 33% better in D1) compared to the validation set (only a 9% improvement) and the test set (10% improvement).

5. Discussion

We demonstrate that a bottleneck tri-cost volume, composed of the sum of absolute differences and two monocular features, can be adopted easily by popular architectures, potentially reducing memory and compute substantially. We further show that Soft CCE loss with local MAP can substantially reduce D1 error rate when compared to training with regression losses, particularly the popular Huber loss.

Our cost-volume experiments demonstrate that monocular features are important for stereo prediction. However, stereo networks should aim to make more efficient use of monocular features than the popular concatenation technique. While our bottleneck tri-cost volume attempts to do so, copying the same 2 reference features across all disparity dimensions remains an inefficient representation.

Our minimum variance, unbiased Soft encoding is the ideal distribution that may be learned through regression losses for a given ground truth. Soft encoding is an intuitive representation for pixel matching in non-occluded regions, where a pixel may be spread across two adjacent pixels in the secondary view. However, in occluded regions, Soft encoding makes less sense. A different loss, such as an ordinal loss, on occluded pixels could improve training, though we leave this as an avenue for future research.

Training with Soft CCE and generating the bottleneck tri-cost volume is straightforward, requiring little custom code. This makes our proposed changes easy to incorporate in current and upcoming stereo networks. To help developers and to demonstrate repeatability, we will release PSM-BTC Net for PyTorch as well as the weights for our PSM-BTC models.

Acknowledgement

This work was sponsored by a gift from Light. The authors would like to thank Dr. Sumit Chawla for valuable discussions and guidance.

References

[1] Yuan Tian, Tao Guan, and Cheng Wang. Real-time occlusion handling in augmented reality based on an object tracking approach. *Sensors (Basel)*, 10(4):2885–900, Mar 2010.

[2] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. *CoRR*, abs/1906.06310, 2019.

[3] Ning-Hsu Wang, Bolivar Solarte, Yi-Hsuan Tsai, Wei-Chen Chiu, and Min Sun. 360sd-net: 360° stereo depth estimation with learnable cost volume. *CoRR*, abs/1911.04460, 2019.

[4] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, Xiaoshuai Sun, and Wenxiu Sun. Toward 3d object reconstruction from stereo images. *CoRR*, abs/1910.08223, 2019.

[5] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. *CoRR*, abs/1803.08669, 2018.

[6] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. *CoRR*, abs/1903.04025, 2019.

[7] Feihu Zhang, Victor Adrian Prisacariu, Ruigang Yang, and Philip H. S. Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. *CoRR*, abs/1904.06587, 2019.

[8] Zhibo Rao, Mingyi He, Yuchao Dai, Zhidong Zhu, Bo Li, and Renjie He. Nlca-net: a non-local context attention network for stereo matching. *APSIPA Transactions on Signal and Information Processing*, 9:e18, 2020.

[9] Xianzhi Du, Mostafa El-Khamy, and Jungwon Lee. Amnet: Deep atrous multiscale stereo disparity estimation networks. *CoRR*, abs/1904.09099, 2019.

[10] Youmin Zhang, Yimin Chen, Xiao Bai, Jun Zhou, Kun Yu, Zhiwei Li, and Kuiyuan Yang. Adaptive unimodal cost volume filtering for deep stereo matching. *CoRR*, abs/1909.03751, 2019.

[11] Kang Zhang, Yuqiang Fang, Dongbo Min, Lifeng Sun, Shiqiang Yang, Shuicheng Yan, and Qi Tian. Cross-scale cost aggregation for stereo matching. *CoRR*, abs/1403.0316, 2014.

[12] Vladimir Tankovich, Christian Häne, Sean Ryan Fanello, Yinda Zhang, Shahram Izadi, and Sofien Bouaziz. Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. *CoRR*, abs/2007.12140, 2020.

[13] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. *CoRR*, abs/1703.04309, 2017.

[14] Sameh Khamis, Sean Ryan Fanello, Christoph Rhemann, Adarsh Kowdle, Julien P. C. Valentin, and Shahram Izadi. Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. *CoRR*, abs/1807.08865, 2018.

[15] Yan Wang, Zihang Lai, Gao Huang, Brian H. Wang, Laurens van der Maaten, Mark Campbell, and Kilian Q. Weinberger. Anytime stereo image depth estimation on mobile devices. *CoRR*, abs/1810.11408, 2018.

[16] Pier Luigi Dovesi, Matteo Poggi, Lorenzo Andraghetti, Miquel Martí, Hedvig Kjellström, Alessandro Pieropan, and Stefano Mattoccia. Real-time semantic stereo matching. *CoRR*, abs/1910.00541, 2019.

[17] Divyansh Garg, Yan Wang, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. Wasserstein distances for stereo disparity estimation. *CoRR*, abs/2007.03085, 2020.

[18] Stepan Tulyakov, Anton Ivanov, and François Fleuret. Practical deep stereo (PDS): toward applications-friendly deep stereo matching. *CoRR*, abs/1806.01677, 2018.

[19] Yang Chen, Zongqing Lu, Xuechen Zhang, Lei Chen, and Qingmin Liao. Noise-sampling cross entropy loss: Improving disparity regression via cost volume aware regularizer. *CoRR*, abs/2005.08806, 2020.

[20] H. Lee and Y. Shin. Real-time stereo matching network with high accuracy. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4280–4284, 2019.

[21] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *CoRR*, abs/1810.02695, 2018.

[22] Gengshan Yang and Deva Ramanan. Upgrading optical flow to 3d scene flow through optical expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[23] Haoyu Ren, Aman Raj, Mostafa El-Khamy, and Jungwon Lee. Suw-learn: Joint supervised, unsupervised, weakly supervised deep learning for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[24] Nikolai Smolyanskiy, Alexey Kamenev, and Stan Birchfield. On the importance of stereo for accurate depth estimation: An efficient semi-supervised deep neural network approach. *CoRR*, abs/1803.09719, 2018.

[25] Anurag Ranjan, Joel Janai, Andreas Geiger, and Michael J. Black. Attacking optical flow. *CoRR*, abs/1910.10053, 2019.

[26] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. *CoRR*, abs/1812.06264, 2018.

[27] Alessio Tonioni, Fabio Tosi, Matteo Poggi, Stefano Mattoccia, and Luigi di Stefano. Real-time self-adaptive deep stereo. *CoRR*, abs/1810.05424, 2018.

[28] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. *CoRR*, abs/1512.02134, 2015.

[29] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. *CoRR*, abs/1709.02371, 2017.

[30] W. Luo, A. G. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5695–5703, 2016.