2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2017)
November 16 - 18, 2017, Daegu, Korea

A Deep Neural Network Approach to Fusing Vision and Heteroscedastic Motion Estimates for Low-SWaP Robotic Applications

E. Jared Shamwell[1], William D. Nothwang[2], Donald Perlis[3]

***Abstract*— Due both to the speed and quality of their sensors and restrictive on-board computational capabilities, current state-of-the-art (SOA) size, weight, and power (SWaP) constrained autonomous robotic systems are limited in their abilities to sample, fuse, and analyze sensory data for state estimation. Aimed at improving SWaP-constrained robotic state estimation, we present Multi-Hypothesis DeepEfference (MHDE) - an unsupervised, deep convolutional-deconvolutional sensor fusion network that learns to intelligently combine noisy heterogeneous sensor data to predict several probable hypotheses for the dense, pixel-level correspondence between a source image and an unseen target image. This new multi-hypothesis formulation of our previous architecture, DeepEfference [1], has been augmented to handle dynamic heteroscedastic sensor and motion noise and computes hypothesis image mappings and predictions at 150-400 Hz depending on the number of hypotheses being generated. MHDE fuses noisy, heterogeneous sensory inputs using two parallel architectural pathways and n (1, 2, 4, or 8 in this work) multi-hypothesis generation subpathways to generate n pixel-level predictions and correspondences between source and target images. We evaluated MHDE on the KITTI Odometry dataset [2] and benchmarked it against DeepEfference [1] and DeepMatching [3] by mean pixel error and runtime. MHDE with 8 hypotheses outperformed DeepEfference in root mean squared (RMSE) pixel error by 103% in the maximum heteroscedastic noise condition and by 18% in the noise-free condition. MHDE with 8 hypotheses was over $5,000\%$ faster than DeepMatching with only a 3% increase in RMSE.**

I. Introduction

The sensing and processing pipelines of autonomous and semi-autonomous robotic systems pose a fundamental limit on how fast these systems may safely travel through an environment. For example, when moving at 20 m/s, a 30 Hz sensor-derived state estimate update rate means that a given robot will travel 0.66 meters between state updates. While traveling those 0.66 meters, the robot will effectively be blind to any unexpected changes in the environment (e.g., a tree branch blown by a wind gust or an unexpectedly opened door). As a result, current size, weight, and power (SWaP) constrained autonomous and semi-autonomous robotic systems are forced to move very slowly through their environments.

The slow operational speeds of SWaP-constrained autonomous systems are especially pronounced for mobile robots operating in dynamic, gps-/communications-denied environments where safe navigation must be performed only with on-board sensors and computational resources. For unmanned aerial vehicles (UAVs), navigation is typically performed through a fusion of visual odometry (VO) estimates, inertial measurements, and simplified predictive linear motion models in a Kalman filter framework. These SWaP-constrained VO-pipelines force the use of lightweight feature matching approaches for visual correspondence that are out-performed by computationally heavier SOA approaches. For example, the visual matching algorithm DeepMatching has enabled SOA matching and optical flow [4] but the correspondence-finding step alone can require from 16 seconds to 6.3 minutes per RGB image pair depending on the parameter regime used for matching [3]. For real-time operation on SWaP-constrained systems, correspondence must be computed orders of magnitude faster (e.g., a minimum of 33 ms per matching pair for a 30 FPS camera commonly used for SWaP-constrained robotic applications).

We argue that contextual information can greatly reduce the computational burden for image correspondence approaches and enable both higher-quality and lower-latency state estimation. One way to provide context is by fusing measurements from multiple sensory modalities. However, intelligently integrating multimodal information into low-level sensory processing pipelines remains challenging, especially in the case of SWaP-constrained robotic systems.

We have previously shown that our architecture DeepEfference [1] can efficiently fuse visual information with motion-related information to greatly increase runtime performance ($20,000\%$) with minimal performance degradation (12%) for dense image correspondence matching. However, in our previous work, we used motion estimates as inputs to DeepEfference that were accurate to within approximately 10 cm of actual pose. In the real-world, systems will rarely have access to comparatively clean signals. Additionally, real noise sources are often heteroscedastic and input-dependent.

With the original DeepEfference's fast runtime, we saw the possibility of generating many different hypothetical outputs for each input image and then selecting the most accurate at execution time. By learning how to produce n image reconstruction predictions, the DeepEfference architecture could be expanded to better handle real-world noise sources.

In this work, we introduce Multi-Hypothesis DeepEf-

*This work was supported by the US Army Research Laboratory

[1]E. Jared Shamwell is a doctoral candidate in the Neuroscience and Cognitive Science Program at the University of Maryland, College Park and a research scientist with GTS stationed at the US Army Research Laboratory, Adelphi, MD 20783. earl.j.shamwell.ctr@mail.mil; ejsham@umd.edu

[2]William D. Nothwang, PhD is the Branch Chief (a) of the Micro and Nano Devices and Materials Branch in the Sensors and Electron Devices Directorate at the US Army Research Laboratory, Adelphi, MD 20783. william.d.nothwang.civ@mail.mil

[3]Donald Perlis, PhD is a Professor of Computer Science at the University of Maryland, College Park, MD 20742. perlis@umd.edu

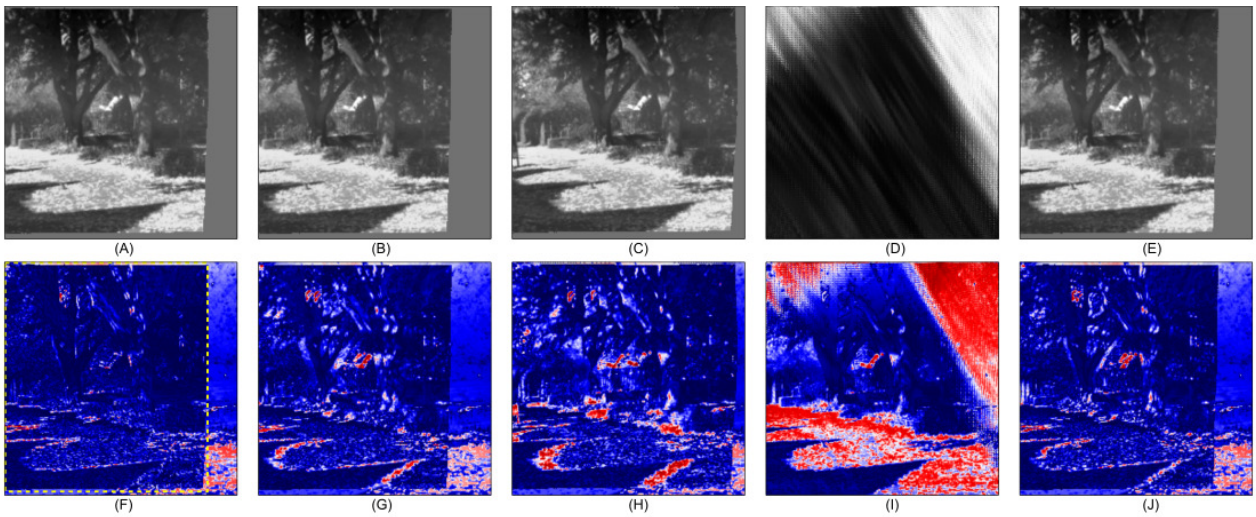


Fig. 1: Sample MHDE outputs from different hypothesis pathways. A-E: MHDE ouputs from 5 pathways. D shows the output from an inactive pathway (i.e. a pathway that the network did not optimize). F-E: Reconstruction error for the hypotheses shown in A-E. From F we can see that the reconstruction shown in A had the lowest error (yellow-dashed box).

ference (MHDE) which is an extension of DeepEfference [1] that mitigates performance impacts of noisy motion estimates. A side-effect of this multi-hypothesis approach is enhanced performance even in the absence of added noise that achieves a mean pixel error within 3% of SOA approaches with an over $5,000\%$ decrease in runtime. By learning how to generate multiple hypothetical outputs, MHDE can effectively sample the space of possible image transformations. This is enabled by a multi-pathway network architecture and novel loss rule that enables the network to explicitly learn multiple, independent network pathways.

The remainder of the paper is organized as follows: Section II describes the background and motivations for MHDE; Section III outlines our deep network approach to fusing noisy heterogeneous sensory inputs and describes the MHDE architecture; Section IV outlines our experimental and evaluation approaches; Section V presents our experimental results; Section VI discusses the results from Section V; and Section VII offers a summary, concluding thoughts, and directions for future work.

II. Background

A. Visual Odometry and Multi-Sensor Fusion

In VO as well as many other vision tasks such as motion understanding and stereopsis, a key challenge is discovering quantitative relationships between temporally or spatially adjacent images. Within the last decade, bio-plausible approaches for the visual task of object recognition have set new benchmarks and are now the defacto standard. We agree strongly with Memisevic that bio-plausible, local filtering-based approaches similarly hold promise for the correspondence problem [5].

A known failure mode for visual odometry (VO) is in highly dynamic scenes. Most VO algorithms are subject to the static scene assumption whereby additional error is introduced when independently moving points in the scene move inconsistently with their dependently moving neighbors.

Feedback outlier detection approaches based on algorithms such as RANSAC [6] seek to discover the most likely motion that has caused a given transform. However an unconstrained key-point match between two images across a large temporal window and spatial extent is at least exponentially complex [7]. By fusing sensor information from separate modalities, we can effectively constrain the matching process.

Constraining the matching process to be consistent with a narrow range of transforms gleamed from another modality can lead to increased VO performance relative to computational requirements and processing time. Previous work has applied extra-visual feedback signals from IMUs or GPS [8], [9] to constrain the matching process. Simple motion models [10], [11] have also been used to predict future images based on previously observed image motion. These approaches have been extended to use quadratic motion models [12] which showed improved performance in specific environments (e.g., on flat roads). However, these models implicitly sacrifice responsiveness as they wait for changes in an underlying sensory distribution rather than detecting dominant motion from a separate extra-visual modality.

B. Deep Spatial Transformations

The correspondence problem describes the challenge of determining how the pixels in one image spatially correspond to the pixels in another image. Traditionally, the correspondence problem has been tackled with closed-form, analytical approaches (see [13] for a review) but recently, deep, bio-inspired, solutions have also begun to show promise. These deep approaches solve the correspondence problem by learning to estimate the 3D spatial transformations between image pairs.

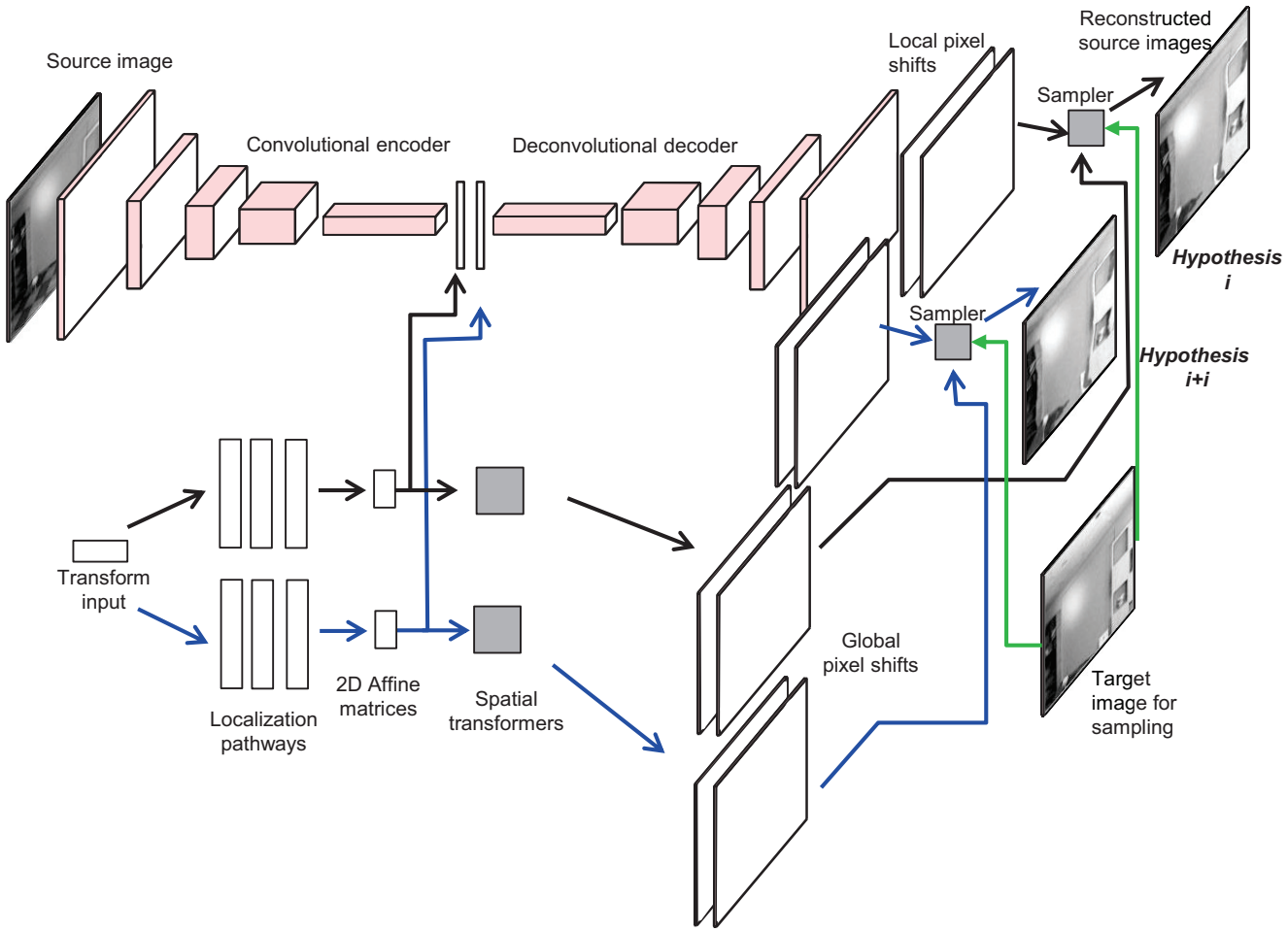


Fig. 2: MHDE network diagram with two hypotheses shown for brevity. We experimented with up to 8 hypotheses in this work.

In computer vision, siamese-like deep network architectures such as those based on multiplicative interactions have been used successfully for relationship learning between images [5], [14]–[18]. However,there are two problems with these and other deep approaches (e.g. the DeepMatching [3], [4] algorithm described earlier) to image transformation learning.

First, these approaches require expensive computation on both initial and target images. They employ siamese architectures that require parameter-heavy learning and expensive computations to be performed on both source and target images. For SWaP-constrained robots, the number of computational operations required by these siamese networks must be significantly reduced. Approaches such as L1 and group lasso-based pruning [19]–[21] offer potential mechanisms to reduce the size of networks but fundamentally still require extensive computation on both source and target images.

Second, these approaches do not provide a mechanism to include extra information from another modality as a motion prior while maintaining end-to-end trainability. For robotic applications, heterogeneous sensor information is often available that can be leveraged and may allow for reduced computational constraints and increased performance (see Section II-C).

C. Extra-Modal Motion Estimates and Heteroscedastic Noise

Unlike algorithms in pure computer vision domains, algorithms intended for robotic applications need not rely solely on vision. For example, when estimating a robotic system's egomotion by tracking changes in feature point locations on a robot's camera's imaging plane, additional non-visual motion estimates can be fused with visual information(i.e. to bias or serve as a motion prior) to improve egomotion estimation.

On real-world systems, additional non-visual motion estimates could be derived from measurements taken from IMUs, GPS, LIDARs, ultrasonic ranging sensors, or the actual input motor commands given to the system. Furthermore, motor errors exhibit heteroscedastic noise properties where larger movements generate larger sources of noise [22]. Any approach that seeks to leverage extra-modal motion estimates needs to be robust to real-world heteroscedastic noise.

III. Approach

MHDE is an unsupervised deep heterogeneous neural network that employs multiple separable pathways to fuse noisy, heterogeneous sensory information and predict how source images correspond to unseen target images.

MHDE effectively reverses the prediction pipeline - rather than using the previous image to reconstruct the future image, it uses the target image to reconstruct the source image. The network receives a noisy estimate of the change in 3D camera position between source and target frame acquisitions and learns:

1) 2D affine transformation parameters that are applied as a global spatial transform; and
2) Local, pixel-level shifts that encapsulate aberrations due to varied scene depth, non-rigid scene objects, etc.

The affine transformations and localized shifts are learned and applied via two interconnected architectural pathways: one for determining global 2x3 affine 2D transformation matrices, and a second encoder-decoder pathway that predicts localized, pixel-level shifts that are not captured by the global, approximated 2D affine transformation (see [1] for more information on the DeepEfference architecture).

Unlike the original DeepEfference, MHDE generates several hypothetical reconstructions which enable increased robustness to noisy inputs. Thus, while DeepEfference only has two architectural pathways, MHDE has the same two architectural pathways plus n additional hypothesis generation pathways ($2 - 8$ in this work).

A. Winner-Take-All (WTA) Loss Rule

MHDE generates multiple hypothesis reconstructions to enable robustness to stochastic, heteroscedastic, input noise such as found in the real-world. The previous DeepEfference architecture that generated only a single predicted reconstruction used Euclidean error to train the network by minimizing the loss function

$$L(\theta, I_t, I_s) = \underset{\theta}{\operatorname{argmin}} \|I_r(\theta, I_t) - I_s\|^2 \tag{1}$$

where I_r is an image reconstruction, I_t is the image target, and I_s is the image source being reconstructed.

If instead of generating a single reconstruction I_r, the network generated n reconstructions $I_r^i, i \in N$, the loss rule would need to be expanded to train across all hypothesis pathways in the new network. A naive way to compute error for such a multi-hypothesis network would be to simply sum the Euclidean error from all hypotheses and divide by the total number of hypotheses. Then, the network would be trained by minimizing the loss function

$$L(\theta, I_t, I_s) = \underset{\theta}{\operatorname{argmin}} \frac{\sum_i^N \|I_r^i(\theta, I_t) - I_s\|^2}{N} \tag{2}$$

where I_r^i is a hypothesis image reconstruction and the remaining terms are the same as before.

The naive multi-hypothesis loss rule of Eq. 2 would lead the network to optimize all pathways simultaneously with each update. However, this may not be optimal for increased robustness to noise. Effectively, we desire the network to generate distinct predictive hypotheses by sampling from a noise distribution that the network implicitly learns. For example, consider when the network has perfectly optimized the loss function of Eq. 2:

$$L(\theta, I_t, I_s) = \frac{\sum_i^N \|I_r^i(\theta, I_t) - I_s\|^2}{N} \approx 0 \tag{3}$$

In this case, $\|I_r^i(\theta, I_t) - I_s\|^2 \approx 0, \forall i \in N$ which means that each hypothesis reconstruction $I_r^i(\theta, I_t)$ is approximately equal. As the network is trained and converges to a local minima, loss will affect parameters in each pathway approximately equally and drive outputs from all pathways to a common approximate solution. This is the opposite of what we want from MHDE. Effectively, such a loss rule is equivalent to the standard Euclidean loss rule used in [1] where a single prediction is generated and fails to leverage the multiple outputs that can be generated by MHDE.

To leverage its multiple outputs, we train MHDE using what we call a winner-take-all (WTA) Euclidean loss rule:

$$I_r^*(\theta, I_t) \longleftarrow \underset{i}{\operatorname{argmin}} \|I_r^i(\theta, I_t) - I_s\|^2 \tag{4}$$

$$L(\theta, I_t, I_s) = \|I_r^*(\theta, I_t) - I_s\|^2 \tag{5}$$

where I_r^* is the lowest error hypothesis. Loss is then only computed for this one hypothesis and error is backpropagated only to parameters in that one pathway. Now, only parameters that contributed to the winning hypothesis are updated and the remaining parameters are left untouched.

B. Pathway 1: Global Spatial Transformer

Spatial transformer (ST) modules [23] apply parametrized geometric transformations to feature-maps (either data inputs or intermediate outputs) in deep networks. The parameters for these transformations (2D affine transformations in our case) can be directly provided to the network as input or can be learned and optimized alongside the other network parameters (e.g., network weights and biases).

MHDE was provided with estimates of the true 3D transformation between source and target images (δx, δy, δz, $\delta\alpha$, $\delta\beta$, $\delta\gamma$). Note, however, that the visual input to MHDE was a single grayscale source image without any depth information. Even if the provided 3D transformation was noise-free and perfectly accurate, it is not possible to analytically perform a 3D warp (assuming translation) on a 2D image due to unknown scene depth at each pixel location. Thus, MHDE approximated 3D warps as 2D affine transformations through a linear-nonlinear optimization using four fully-connected layers, each followed by an additional rectified linear unit (ReLU) [24] non-linearity layer.

We modified the standard ST module in tensorflow [25] by splitting the layer into two layers - one to perform the affine transformation on grids of source pixel locations (x^s, y^s) and output target pixel coordinates (x^t, y^t) and a second layer to

perform bilinear sampling given pixel coordinates and an image to sample from.

Although the sampling component of our ST module takes an input image as input, no learn-able parameters are based on input image content and thus our global pathway is a function only of the input transformation estimate and is image content-independent.

C. Pathway 2: Local, Pixel-Level Shifter

The pixel-level encoder/decoder pathway refines the ST estimate from the first pathway and provides localized estimates of pixel movement to account for depth, non-rigidity, etc.

We implemented this pathway as a convolutional-deconvolutional encoder-decoder. First, the convolutional encoder compresses a source image through a cascade of convolutional filtering operations. The output of the convolutional encoder is concatenated with intermediate outputs from the fully-connected layers from the first, global pathway (the black and blue vertical lines in the center of Fig. 2). This concatenated representation is then expanded using a deconvolutional decoder to generate n pairs of $(x^{t'}, y^{t'})$ pixel locations that are summed with the target pixel coordinates (x^t, y^t) from the global pathway before bilinear sampling (see [1] for more details).

IV. Experimental Methods

We conducted experiments with MHDE using four different noise conditions and four different architectures. All architectures were based on DeepEfference [1] and implemented both global pathway and local pathways. MHDE was evaluated on the KITTI Odometry dataset [2] and results were benchmarked against correspondence matching results from the SOA DeepMatching approach [3] (see [1] and the Appendix for more information).

We experimented with four noise conditions where α was 0.0 , 0.1, 0.25, or 0.5. We trained networks with 1, 2, 4 or 8 hypothesis generation pathways. For each noise and and hypothesis combination, we trained three networks for a total of 48 different networks.

A. Noise

As shown in Fig. 3, we simulated real-world noise conditions by applying heteroscedastic noise to each transform input. For each transform $T = (\delta x, \delta y, \delta z, \delta\alpha, \delta\beta, \delta\gamma)$, we introduced heteroscedastic noise to create network input T^* according to:

$$T^* = T + \mathcal{N}(0, \alpha\sqrt{T}) \tag{6}$$

where α was a constant modifier that was either 0.0, 0.1, 0.25, or 0.5.

B. Evaluation

We evaluated MHDE by measuring the mean pixel error of MHDE projections of DeepMatching keypoints from source images to target images. The projection errors for each method compared to groundtruth projections were used to determine mean pixel errors for each method (see [1] and Appendix A. for a more thorough explanation of the experimental evaluation).

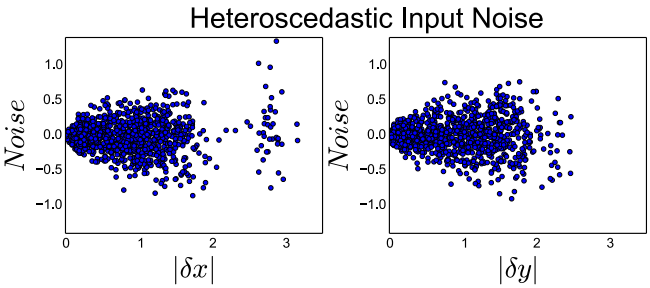


Fig. 3: Heteroscedastic noise as a function of transform magnitude for the X and Y components of the transform input over the test set for a network with a noise parameter $\alpha = 0.25$.

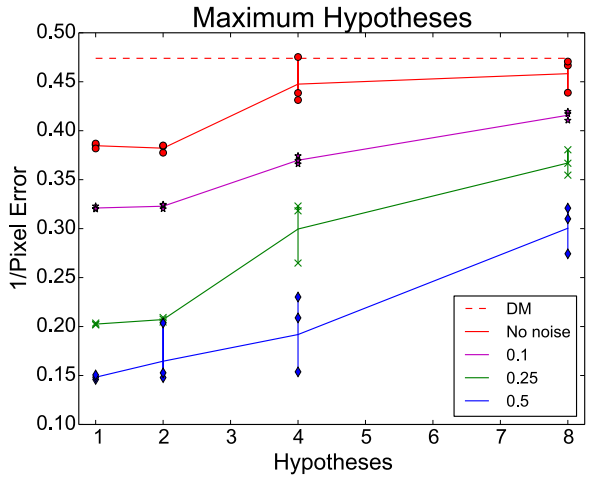


Fig. 4: Inverse mean pixel error (higher is better) for several noise conditions produced by MHDE networks trained to generate 1, 2, 4, or 8 maximum hypotheses. Dashed line is DM (SOA) error.

C. Training

We trained MHDE for $200,000$ iterations on KITTI Odometry scenes $1 - 11$ for all experiments. We used the Adam solver with batch size=32, momentum1=0.9, momentum2=0.99, gamma=0.5, learning rate=$1e-4$, and an exponential learning rate policy for all experiments. All networks were trained using our modified WTA loss rule. All experiments were performed with a Nvidia Titan X GPU and Tensorflow (see [1] and the Appendix for a more thorough explanation of training procedures).

V. Results

Fig. 4 shows the performance of MHDE with various maximum hypotheses compared to DM. A network's maximum hypotheses is the maximum number of hypothesis generation pathways a given network was allowed to learn.

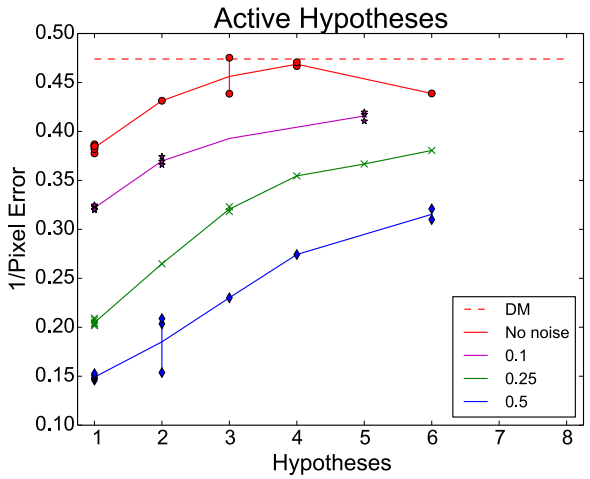


Fig. 5: Inverse mean pixel error (higher is better) for several noise conditions produced by MHDE networks. Results are from the same networks shown in Fig. 4 but are instead plotted as a function of active pathways learned by each network.

Because of our WTA loss rule, this does not mean that the network effectively learned how to use all pathways. For example in Fig. 6d, a network with four maximum hypotheses predominantly trained and used a single pathway.

This can also be seen in Fig. 5 where the same results from Fig. 4 are plotted as a function of the total active hypotheses. Active hypotheses are considered hypothesis pathways that performed better than all other pathways for at least one testing expemplar (for reference, Fig. 1(d) shows the network output of an inactive pathway).

There is positive relationship between performance and both maximum hypotheses and active hypotheses. This is true for all noise conditions as well as the no-noise condition. We also see that the rate of improvement when moving from one to six active hypotheses is greater for higher noise levels.

Fig. 6 shows the activations by pathway for networks trained with four or eight maximum hypotheses. Surprisingly, regardless of noise conditions, we see no strong relationship between active pathways (pathways that produced the best result for at least one test exemplar) and noise level.

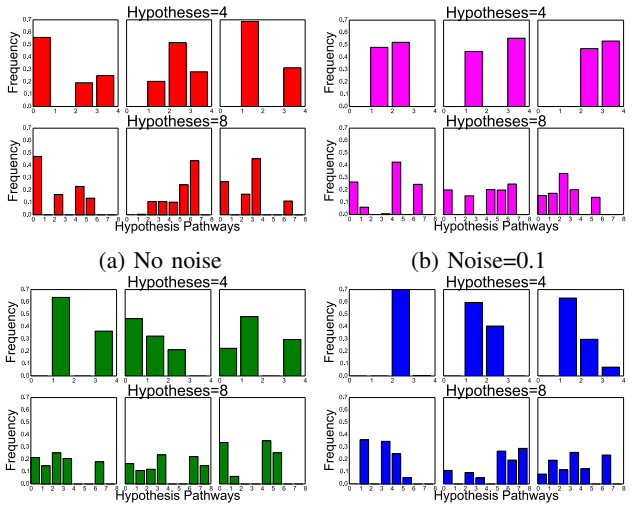
Tab. I details the comparative runtimes between DeepMatching and MHDE with various numbers of hypotheses. MHDE runtime scales linearly with number of hypotheses. Overall, the runtime gains of MHDE compared to DM show that providing a strong prior on camera motion allows for far more computationally efficient image predictions and matchings.

VI. Discussion

We were concerned that MHDE networks might only optimize a single pathway. For example, if one pathway consistently produced the lowest estimate error at the beginning of training, then perhaps only that pathway would be updated and thus the network would not be used to it's

TABLE I: Average runtimes for DeepMatching (DM) and Multi-Hypothesis DeepEfference (MHDE) with equivalent frames per second (FPS)

	# Hypoth.	*Mean*	*StDev.*	*Med.*	*FPS*
DM	N/A	0.4115 s	0.00132	0.407 s	2.4
MHDE	1	0.0024 s	0.00008	0.00238 s	417.4
MHDE	2	0.00303 s	0.00009	0.00302 s	330
MHDE	4	0.00422 s	0.00010	0.00421 s	237.2
MHDE	8	0.00675 s	0.00016	0.00677 s	148.2



(a) No noise (b) Noise=0.1

(c) Noise=0.25 (d) Noise=0.5

Fig. 6: Activation by pathway for the different noise conditions. Only networks with maximum hypotheses of 4 or 8 are shown.

fullest potential. As seen in Fig. 6, this generally was not the case as networks were able to learn to use multiple pathways without intervention outside of the WTA loss rule.

Future work will look at how to include pure sensor measurements (e.g., from an IMU) and how to encourage networks to train and use all available hypothesis pathways. Like it's predecessor, MHDE only uses single grayscale images as inputs. Another possible avenue of research is to use multiple images as input, or an LSTM like architecture to give the network additional temporal context.

One of the more important aspects of this network is that it does not generate images from scratch and instead works mostly in the space of pixel locations rather than pixel intensities. Given that geometry is consistent across image domains even though image content varies, this network architecture is a promising candidate to leverage transfer learning.

While we used noise-corrupted motion estimates derived from ground-truth for the MHDE transform input, IMUs are a possible real-world source for this information. However, IMUs only measure accelerations and thus we speculate that using raw IMU measurements as MHDE inputs will result in poor performance during constant velocity maneuvers. Additional work is needed to determine a suitable real-world

analog for deriving the motion estimates needed by MHDE.

We hope to experiment with this architecture on other visual odometry datasets. Specifically, we seek a larger dataset with a wider range of movements. Without a wide range of movements, we speculate that trained networks will only be able to transfer to new, previously unseen datasets that follow similar movement statistics as the datasets on which they were trained. To overcome many of these limitations, we are currently working to collect a multi-modal dataset with stereo imagery, depth imagery, high-resolution IMU data, action commands, low-level motor-commands, and ground-truth VICON poses. With this dataset, we will be able to better address limitations inherent in the current MHDE architecture.

VII. Conclusion

While increased performance in the noise-free conditions was an unintended consequence of the multi-hypothesis formulation, the central contribution of this work is in the handling of noise-contaminated input data. In summary, we have shown the unsupervised learning of correspondence between static grayscale images in a deep sensorimotor fusion network with noisy sensor data. In this work, we have presented a multi-hypothesis formulation of our previous DeepEfference architecture. MHDE outperformed DE by 103% in RMSE in our maximum noise condition, by 18% in the noise-free condition, and was 181% slower (417 FPS vs 148 FPS). Compared to DM, MHDE was 5192% faster with 8 hypotheses (2.8 FPS vs 148 FPS) and was outperformed by 3% in the noise-free condition with 8 hypotheses and by 57% in the maximum noise condition with 8 hypotheses.

APPENDIX

The following methods are largely reproduced from [1] and included here for completeness.

A. Extended Evaluation

As in [1], we evaluated MHDE on the KITTI Visual Odometry dataset [2]. KITTI is a benchmark dataset for the evaluation of visual odometry and LIDAR-based navigation algorithms. Images in KITTI were captured at 10 Hz from a Volkswagen Passat B6 as it traversed city, residential, road, and campus environments. Groundtruth poses at each camera exposure were provided by an RTK GPS solution and depth is provided with coincident data from a Velodyne laser scanner. All objects in the visual scenes are rigid, thus fulfilling the static scene assumption and allowing for ground truth to be computed from scene depth and camera position.

Predicted pixel correspondence between source and target images was evaluated against groundtruth correspondence and SOA DeepMatching correspondence predictions. With access to scene depth and true camera pose for KITTI, groundtruth pixel shifts were calculated by applying a 3D warp to 3D pixel locations in the source images to generate the expected pixel locations in the target images. We projected each 3D point in the frame of $camera_{t0}$ to the world frame using the derived projection matrix for $camera_{t0}$ and then reprojected these points in the world frame to $camera_{t1}$ using the inverse projection matrix for $camera_{t1}$. Finally, we transformed points in the frame of $camera_{t1}$ to the image plane. This resulted in a correspondence map between pixel locations in $camera_{t0}$ and $camera_{t1}$ for each point where depth was available (e.g., when depth was outside of the Velodyne laser scanner's range).

B. Extended Training Procedures

For training and evaluation, data was separated into train (80%) and test (20%) sets. We used a total of $23,190$ image pairs with 80% $(18,552)$ for training and 20% $(4,638)$ for testing. In all experiments, we randomly selected an image for the source, used the successive image for the target, and subtracted the two 6-DOF camera poses for the transform input. For each image in each dataset, we cropped the middle 224x224 pixel region for network inputs.

References

[1] E. J. Shamwell, W. D. Nothwang, and D. Perlis, "Deepefference: Learning to predict the sensory consequences of action through deep correspondence," in *Development and Learning and Epigenetic Robotics (ICDL), 2017 IEEE International Conference on*. IEEE, Accepted.

[2] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[3] J. Revaud, P. Weinzaepfel, Z. Harchaoui, C. Schmid, J. Revaud, P. Weinzaepfel, Z. Harchaoui, C. S. D. Hi, and C. Schmid, "DeepMatching : Hierarchical Deformable Dense Matching," *International Journal of Computer Vision*, vol. 120, no. 3, pp. 300–323., 2016.

[4] J. Revaud, P. Weinzaepfel, Z. Harchaoui, C. Schmid, J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. S. E. Edge, "EpicFlow : Edge-Preserving Interpolation of Correspondences for Optical Flow," *Cvpr 2015*, 2015.

[5] R. Memisevic, "Learning to relate images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1829–1846, 2013.

[6] B. Kitt, F. Moosmann, and C. Stiller, "Moving on to dynamic environments: Visual odometry using feature classification," *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pp. 5551–5556, 2010.

[7] T. Brox, J. Malik, and C. Bregler, "Large displacement optical flow," *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 41–48, 2009.

[8] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the Mars Exploration Rovers," *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.

[9] M. Agrawal and K. Konolige, "Real-time localization in outdoor environments using stereo vision and inexpensive GPS," *Proceedings - International Conference on Pattern Recognition*, vol. 3, pp. 1063–1068, 2006.

[10] W. Enkelmann, "Obstacle detection by evaluation of optical flow fields from image sequences," *Image and Vision Computing*, vol. 9, no. 3, pp. 160–168, jun 1991.

[11] A. J. Davison, "Real-time Simultaneous Localisation and Mapping with a Single Camera," *Iccv*, vol. 2, pp. 1403–1410, 2003.

[12] G. Lefaix, T. Marchand, and P. Bouthemy, "Motion-based obstacle detection and tracking for car driving assistance," *Object recognition supported by user interaction for service robots*, vol. 4, no. August, pp. 74–77, 2002.

[13] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[14] R. Memisevic and G. Hinton, "Unsupervised Learning of Image Transformations," *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[15] M. A. Ranzato and G. E. Hinton, "Factored 3-Way Restricted Boltzmann Machines For Modeling Natural Images," *Artificial Intelligence*, vol. 9, pp. 621–628, 2010.

[16] R. Memisevic and G. E. Hinton, "Learning to represent spatial transformations with factored higher-order Boltzmann machines." *Neural computation*, vol. 22, no. 6, pp. 1473–1492, 2010.

[17] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6791 LNCS, no. PART 1, pp. 44–51, 2011.

[18] J. J. Kivinen and C. K. I. Williams, "Transformation equivariant Boltzmann machines," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6791 LNCS, no. PART 1, pp. 1–9, 2011.

[19] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[20] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 2074–2082.

[21] S. Anwar, K. Hwang, and W. Sung, "Structured pruning of deep convolutional neural networks," *arXiv preprint arXiv:1512.08571*, 2015.

[22] C. Ciliberto, S. R. Fanello, L. Natale, and G. Metta, "A heteroscedastic approach to independent motion detection for actuated visual sensors," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3907–3913, 2012.

[23] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial Transformer Networks," *Nips*, pp. 1–14, 2015.

[24] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks." in *Aistats*, vol. 15, no. 106, 2011, p. 275.

[25] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org.