DeepEfference: Learning to Predict the Sensory Consequences of Action Through Deep Correspondence

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

***Abstract*— As the human eyeball saccades across the visual scene, humans maintain egocentric visual positional constancy despite retinal motion identical to an egocentric shift of the scene. Characterizing the underlying biological computations enabling visual constancy can inform methods of robotic localization by serving as a model for intelligently integrating complimentary, heterogeneous information. Here we present DeepEfference, a bio-inspired, unsupervised, deep sensorimotor network that learns to predict the sensory consequences of self-generated actions. DeepEfference computes dense image correspondences [1] at over 500 Hz and uses only a single monocular grayscale image and a low-dimensional extra-modal motion estimate as data inputs. Designed for robotic applications, DeepEfference employs multi-level fusion via two parallel pathways to learn dense, pixel-level predictions and correspondences between source and target images. We present quantitative and qualitative results from the SceneNet RGBD [2] and KITTI Odometry [3] datasets and demonstrate an approximate runtime decrease of over 20,000% with only a 12% increase in mean pixel matching error compared to DeepMatching [4] on KITTI Odometry.**

I. INTRODUCTION

For an autonomous agent (be it robotic or organic), understanding how self-produced actions affect the environment is critically important to survival and successful operation in the real-world. Similarly important is an agent's understanding of how its actions affect its sensory perceptions. In the case of visual positional constancy, this corresponds to an agent's ability to separate motion across the retinal plane induced by self-motion (e.g., from a saccade) from motion induced externally (e.g., from a charging predator).

A comparable understanding of the perceptual consequences of self-induced actions could be used by autonomous robots to measure action-based and perceptual anomalies (among others). Take for example the act of turning to the left. In the case of the former, this action should result in an object within the visual field-of-view (e.g., a soda can) shifting to the right on the imaging plane by a commensurate amount. If this shift does not occur, it could mean that the action was not properly performed and that there could be a problem with the system's actuators. For the latter, the same expectation violation might mean that the soda can moved independently (e.g., was blown by the wind or kicked by a passerby) and subsequently, would be a poor choice as a landmark for visual dead-reckoning[1].

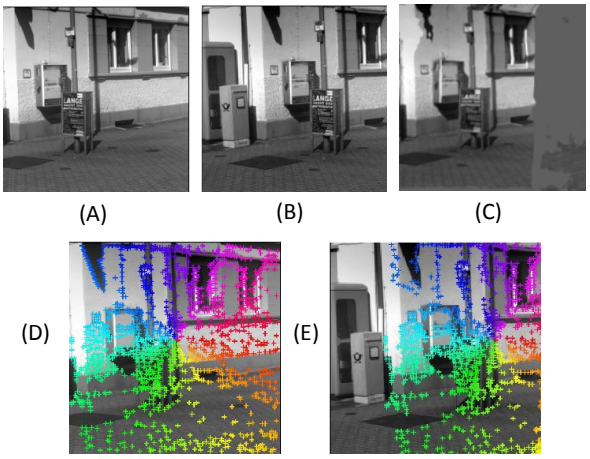

Fig. 1. Sample results from KITTI Odometry. A: Sample source image. B: Sample target image. C: DeepEfference output reconstruction of the source image in A using pixel intensity values sampled from the target image in B. D and E: Source and target images with marked correspondence points computed by DeepEfference.

We argue that biological mechanisms supporting visual constancy contain a rich, egocentric representation of the environment and with appropriate models and computational architectures, these representations can be extracted to enable enhanced robotic visual navigation and localization (e.g., dead-reckoning). Humans maintain perceptual stability and visual constancy despite the 3-4 saccades the human eyeball undergoes per second. While the shift in the projection of the visual world on the retina elicited by a saccade is identical to the shift that alternatively would be elicited by a quick, external shift of the visual world, humans are able to perceptually distinguish between the two conditions and perceive a stable world in the first, and a moving world in the second.

The apparent conundrums of human visual positional constancy can be resolved when considering humans as

*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

[1]While the immediate focus of this paper is on sensorimotor modeling and prediction in the visual domain, this work is situated within a broader class of issues including that of how an agent can respond appropriately to anomalies in a complex world (see [5], [6] for Mauthner Cell anomaly detectors in teleost fish; [7], [8] for EC in the auditory domain for human-robot interaction; [9]–[11] for independent motion detection).

complex, embodied agents with access to information from multiple overlapping sensory modalities including vision, audition, proprioception, 'thought perception' [12], [13], and intentional/motor information [7], [14], [15]. For example, Efference Copy (EC) [14] and the closely related Corollary Discharge (CD) [15] neural theories have long been implicated in the brain's ability to maintain visual positional constancy trans-saccadically. EC and CD posit that early sensory centers access extra-modal information about intended actions to influence subsequent sensory processing by priming early sensory centers with a prior on expected incoming sensory signals.

We drew inspiration from the theories of EC and CD in developing a computational solution to robotic visual localization. Similar to how EC can be used by biological systems to estimate expected sensory information, robotic systems often have access to information with which they can glean an estimate of ego-motion from a separate, non-visual modality. If intelligently integrated, this extra-visual estimate can serve as a prior on expected post-movement visual perceptions and improve visual motion estimates.

Paralleling the biological theory of EC where visual processing centers receive motion/intention-related information to aid in sensory processing, we have designed DeepEfference as an unsupervised, feed-forward, heterogeneous, deep network that computes dense correspondence [1] and performs next-frame prediction at over 500 Hz.

Critical to achieving this update rate, DeepEfference uses monocular images and only processes the source image from each pair. The network learns (x,y) pixel locations of where to sample in the target image to best reconstruct the source image. This translates to learning which pixels in the source image best correspond to the target image, and thus, a correspondence mapping between source and target images.

The remainder of the paper is organized as follows: Section II describes the motivations for this work; Section III outlines the DeepEfference network architecture; Section IV describes the datasets and experiments used for validation; Section V discusses results from the validation experiments; and Section VI offers concluding thoughts and directions for future work.

II. BACKGROUND

A. Deep Approaches to Spatial Transformation Encoding and Learning

Learning spatial transformations and relationships between successive images has been a topic of great interest both in computer vision and robotics and deep, bio-inspired, solutions have already begun to show promise for the correspondence problem (see [1] for a review of the correspondence problem). In computer vision, multiplicative interactions have been used to great success for relationship learning between images [16]–[21]. However, both the initial and transformed image are required as inputs and there is no readily-available means to provide the model extra information from another modality as a motion prior. Both points (but in particular the latter) have implications for the correspondence problem and image relationships for robotics.

These and other deep approaches [4], [22] to spatial transformation encoding rely on siamese-like networks where both source and target images are available and computed on. If we want to deploy deep approaches on SWaP-constrained systems, networks require significant size reductions.

B. Extra-Visual Motion Estimates

For any two visual measurements taken successively, robots often have an independent measurement of self-motion between those two images. These measurements could come from IMUs, GPS, LIDAR, ultrasonic ranging sensor, or input motor commands. When estimating motion based on the movement of feature points on the visual imaging plane, additional non-visual motion estimates could be used as a prior for estimating camera motion. Similar in spirit to this work is [23] where heteroscedastic models were learned for independent motion detection for an actuated camera. However, camera motions were limited to pure rotations, which are not affected by varying depths within a scene. Additionally, [23] was constrained to use Gaussian process models and was not end-to-end trainable.

III. APPROACH

DeepEfference is an unsupervised, deep heterogeneous neural network that learns to predict how source images correspond to unseen (i.e., unprocessed) target images. Rather than learning how to transform each pixel (e.g., via a fully-connected layer), we employ a trainable 2D spatial transformer to impose a global estimate of image motion. Inspired by the Landmark Theory of visual positional constancy [24], DeepEfference carries only a sparse gist of the previous visual scene forward and instead uses the currently perceived image from which to sample.

As shown in Fig. 2, DeepEfference has two interconnected pathways: one for determining the global 2x3 affine 2D transformation matrix, and a second encoder-decoder pathway that predicts local, pixel-level shifts to be applied to the affine-transformed image. The network does not generate images from scratch, but rather learns how to sample from a target image to recreate the initial image. Given a source image and an estimated transform, DeepEfference learns coordinates (x, y) at which to sample in a target image to reconstruct the source image. The result is a correspondence map between pixel locations in the source image and pixels in the target image (see Fig. 1 for example learned correspondences).

A. Training and Loss Rule

DeepEfference is trained to minimize reconstruction errors between a given source image and a reconstruction of that source image generated by selectively sampling from a target image. We compute Euclidean error and use it to train the network via backpropagation. DeepEfference is trained by minimizing the following loss function:

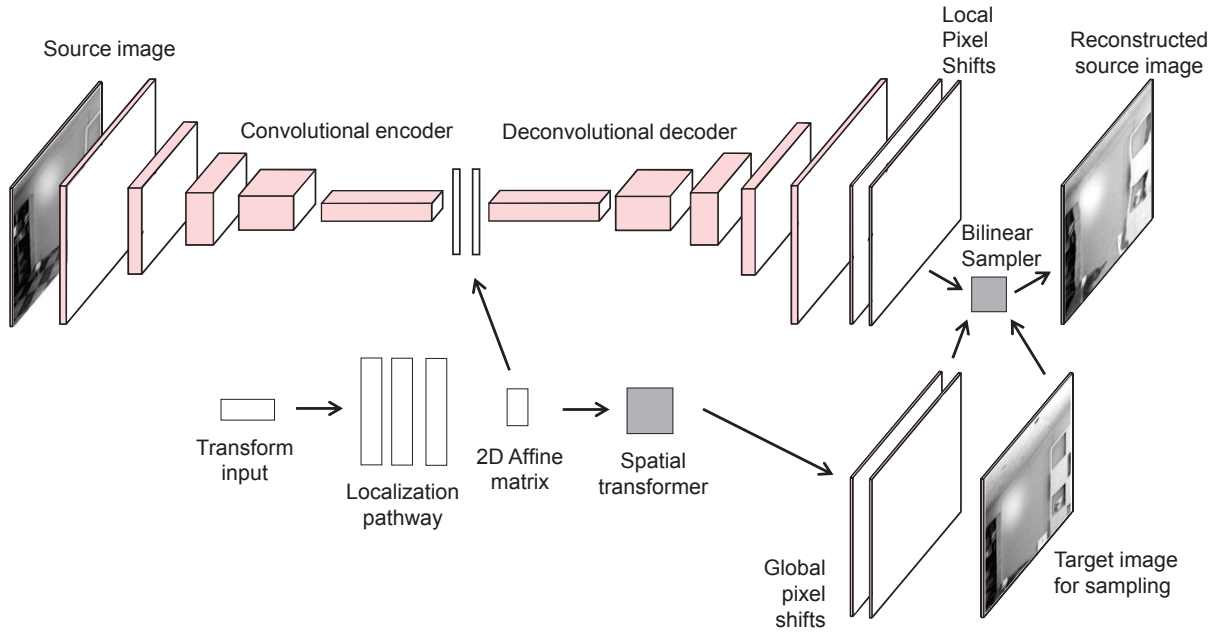

Fig. 2. DeepEfference network diagram showing the linked global and local learners

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

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

B. Pathway 1: Local, Pixel-Level Shifter

DeepEfference's first pathway provides localized, object-level shift information. We implemented this pathway as a convolutional-deconvolutional encoder-decoder. The encoder compresses the source image through a series of convolutional filtering operations and the decoder generates magnitudes of pixel shifts by expanding the compressed convolutional outputs using deconvolutions[2]. We used five convolutional layers followed by five deconvolutional layers. All convolutional and deconvolutional layers used filters of size 3, pad of 1, and stride of 2. The first convolutional layer outputted 32 feature maps and the number of output maps doubled for each subsequent convolutional layer with the fifth and final convolutional layer outputting 512 feature maps. The output sizes of the generative deconvolutional layers were arranged oppositely with the first layer outputting 512 maps and the final layer outputting 32 maps.

The local, pixel-level shifts of DeepEfference are similar to mappings learned by the recent view synthesis method [25] that has been used to render new, unseen views of objects and scenes. Besides different network structures and inclusion of a global spatial transformer module, the largest difference between their method and our own is that rather than learning to generate novel viewpoints of objects or scenes, we learn how to reconstruct a source image using pixel locations in a target image.

[2]We use the term 'deconvolution' as is common in the deep learning literature but the operation we use is more properly referred to as a transposed convolution

C. Pathway 2: Global Spatial Transformer

While the output of the first encoder/decoder pathway provides estimates of localized, pixel-level movement to account for depth, non-rigidity, etc., the second spatial transformer (ST) pathway provides an estimate of the global transformation between source and target images.

ST modules [26] enable parametrized geometric transformations to be applied to inputs or intermediate feature-maps in deep networks. While the parameters for the geometric transformation can either be learned or provided to the network as an input, we provided the network with an estimate of the true 3D transformation between source and target images (δx, δy, δz, $\delta\alpha$, $\delta\beta$, $\delta\gamma$) and used four fully-connected layers (each followed by a rectified linear unit (ReLU) [27]) to approximate the true linear 3D warp matrix as a 2D affine transformation.

However, the failure of a 2D ST-only approach is seen with translational camera movements in scenes with varying depth. Following a camera translation, the new location of an object in the image frame will depend on its distance from the camera: objects that are closer to the camera exhibit greater displacements on the imaging plane compared to objects further from the camera. A purely 2D affine transformation in the absence of depth cannot accurately warp an image with varied scene depths and thus the localization pathway can at best learn parameters that correspond to a dominant plane of a fixed depth.

As we show, ST modules can be used to efficiently embed action or motor information in a standard deep network that may then be trained end-to-end with back-propagation.

These motor-related signals can be derived from high-level actions, referent signals to PID controllers, GPS, or IMU measurements.

We implemented an ST module in Caffe [28] using Nvidia CUDA Deep Neural Network library (cuDNN) primitives. We created two layers, one to perform the affine transformation and output target coordinates and a second layer that performs the bilinear sampling given coordinates and an image. Three fully connected layers take the input estimate 3D camera pose transformation and generate a 2x3 2D affine transformation matrix for the spatial transformer module. Although the sampling component of our ST module takes an input image as input, no learnable parameters are based on image content and thus our global pathway is a function only of the input transformation estimate and is image content-independent.

IV. EXPERIMENTS

We primarily experimented with two different network architectures. The first architecture, LightEfference, only used the first global pathway. The second architecture, DeepEfference, implemented both the global pathway and the local pathway. LightEfference and DeepEfference were evaluated on the SceneNet RGB-D [2] and KITTI Odometry [3] datasets and compared against correspondence matching results from the SOA DeepMatching approach [4].

We also experimented with a third architecture that only used the local pathway. However, networks trained with this architecture failed to converge or decrease network loss (see Section VI for additional discussion on this point).

SceneNet RGB-D [2] is a dataset of 5 million photo-realistically rendered images from a dynamically moving camera in a total of 15 different scenes. Images in SceneNet are rendered at 1 Hz and groundtruth camera pose and depth are provided at each camera exposure. 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 (described in Section IV-A).

The KITTI Visual Odometry dataset [3] 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 are provided by an RTK GPS solution and depth is provided with coincident data from a Velodyne laser scanner. Groundtruth pixel projections were calculated just as for SceneNet.

A. Experimental Methods

For SceneNet and KITTI, data was separated into train (80%) and test (20%) sets. For SceneNet RGB-D, we used a total of $44,850$ image pairs with 80% $(35,880)$ for training and 20% $(8,970)$ for testing. For KITTI, 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.

Predicted pixel correspondences between source and target images were evaluated against groundtruth correspondence and SOA DeepMatching correspondence predictions. With access to scene depth and true camera pose for both KITTI and SceneNet, 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., where ray tracing did not go to infinity in the case of SceneNet or depth was outside of the Velodyne laser scanner's range for KITTI).

We evaluated DeepEfference and LightEfference using keypoints generated using the feature points detected by DeepMatching and from the accelerated segment test (FAST) feature detector [29]. For each type of feature, we measured how the keypoints detected in the source images were projected into the target images. The projection errors were compared to groundtruth projections and were used to determine mean pixel errors for each method.

We trained DeepEfference and LightEfference for $500,000$ iterations on KITTI Odometry scenes $1-11$ and a subset of SceneNet RGBD (10 randomly selected trajectories of 300 image pairs for each of the 15 different scene types). We used the Adam solver with batch size=32, momentum=0.9, momentum=0.99, gamma=0.5, and a step learning rate policy of $100,000$ for all experiments. We used a Euclidean loss rule to train all networks. All experiments were performed with a Nvidia Titan X GPU and the Caffe deep learning framework [28].

V. RESULTS

Tab. V details the comparative runtimes between DeepMatching, LightEfference, and DeepEfference[3]. Fig. 3 and Tab. II detail the predictive error for LightEfference, DeepEfference, and DeepMatching on the KITTI Odometry and SceneNet RGBD datasets. Using keypoints generated by DeepMatching on KITTI, DeepEfference shows a 1,100% performance increase in mean pixel error over LightEfference (significant with $t(2.94\text{e}5) = 60.01$, $p < 1\text{e}{-5}$)[4]

[3]We used a CPU version of DeepMatching for these comparisons. The latest available version of the GPU implementation of DeepMatching took over 7 seconds per image to run on our workstation on a 256x256 image so we elected to use the faster CPU version for all experiments

[4]The distributions of pixel errors appeared non-uniform but as we had greater than 200,000 samples to test in each condition, we elected to include t-test analysis. We used Welch's t-test where the degrees of freedom are approximated by Satterthwaite's method.

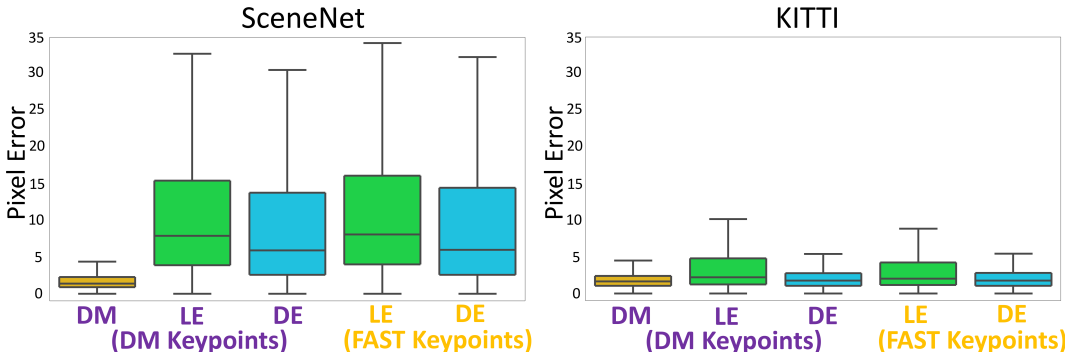

Fig. 3. Pixel error boxplots for DM, LE, and DE using DM and FAST keypoints. Y-axis is actual mean pixel error. Middle lines are the medians and whiskers indicate 1.5 interquartile of the lower and upper quartiles. Note outliers are not shown for clarity and instead minimum and maximums are presented in the table below.

while DeepMatching showed a 12% increase over DeepEfference (significant with $t(5.33e5) = 31.33$, $p < 1e{-}5$). When using FAST keypoints, DeepEfference outperformed LightEfference by 1,200% on KITTI odometry (significant with $t(6.18e5) = 87.87$, $p < 1e{-}5$). However, in runtime performance, LightEfference was 447% faster than DeepEfference, and DeepEfference was over 23,000% faster than DeepMatching.

TABLE I

AVERAGE RUNTIMES FOR DM, LE, AND DE

	Mean	*StDev.*	*Med.*	*FPS*
DM	0.35225 sec.	0.0094525 sec.	0.351561	2.8
LE	0.000332 sec.	2.95788e-05 sec.	0.000318	3012
DE	***0.0014864*** sec.	***2.03606e-05*** sec.	***0.00148484***	***672***

TABLE II

PIXEL ERRORS FOR DM, LE, AND DE ON KITTI AND SCENENET

	KITTI DeepMatching Keypoints				
	Mean	*StDev.*	*Med.*	*Min*	*Max*
DeepMatching	2.1	2.6	1.6	0.0	187.1
LightEfference	29.2	242.3	2.2	0.0	4661.4
DeepEfference	***2.3***	***3.7***	***1.7***	***0.0***	***268.2***
	KITTI FAST Keypoints				
	Mean	*StDev.*	*Med.*	*Min*	*Max*
LightEfference	31.6	261.3	2.0	0.0	4791.2
DeepEfference	***2.4***	***3.7***	***1.7***	***0.0***	***251.3***
	SN DeepMatching Keypoints				
	Mean	*StDev.*	*Med.*	*Min*	*Max*
DeepMatching	3.3	16.2	1.3	0.0	2620.9
LightEfference	12.9	19.8	7.8	0.0	2596.1
DeepEfference	***11.5***	***19.2***	***5.9***	***0.0***	***2691.8***
	SN FAST Keypoints				
	Mean	*StDev.*	*Med.*	*Min*	*Max*
LightEfference	14.8	28.1	8.0	0.0	2957.8
DeepEfference	***13.1***	***26.2***	***5.9***	***0.0***	***3055.4***

The performance gap between LightEfference and DeepEfference narrowed on the SceneNet dataset. DeepEfference outperformed LightEfference by 11% (significant with $t(2.78e6) = 58.65$, $p < 1e{-}5$) and was outperformed by DeepMatching by 240% (significant with $t(2.70e6) = 382.56$, $p < 1e{-}5$) using DeepMatching generated keypoints. With FAST keypoints, DeepEfference performed only 13% better than LightEfference (significant with $t(4.76e6) = 68.64$, $p < 1e{-}5$).

The performance differences between DeepEfference and DeepMatching may be influenced by the differences in movement statistics between the SceneNet and KITTI datasets. For SceneNet, rotational speeds[5] (in deg/s) were typically much larger (mean=5.14, std=9.82, median=2.843, min=0.039, max=177.84) while translational speeds (in m/s) were smaller (mean=0.16, std=0.091, median=0.14, min=0.003, max=0.66). Motions in KITTI followed reversed distributions where rotational speeds (in deg/s) were typically small (mean=0.997, std=6.65, median=0.086, min=0.0001, max=85.72) while translational speeds (in m/s) were much larger (mean=9.22, std=4.20, median=9.04, min=0.005, max=26.41). The larger variety of movements in SceneNet may have proved too difficult for the current version of DeepEfference to learn a consistent motion model. Future work will include experiments on deeper versions of DeepEfference.

The similar performance of LightEfference and DeepEfference on SceneNet may have been influenced by the depth of objects in the dataset. Image scenes in KITTI had both larger and more varied depths (mean=12.97, std=10.005, median=9.46, min=5.00, max=79.99) compared to SceneNet (mean=3.92, std=2.45, median=3.40, min=0.00, max=19.99). This might explain why LightEfference's average pixel prediction error was within 11-13% of DeepEfference as errors from translations of objects at different depths would have been smaller. For example, the green boxes in Fig. 4 highlight an unusual instance in SceneNet where LightEfference was unable to rectify the depth differences between the foreground features and background features.

The poorer performance of DeepEfference and LightEfference on SceneNet may also be due to shifts between

[5]SceneNet only contains renders of 1 in every 25 frames so these quantities are based on the differences in positions between successively rendered frames

successive images resulting in objects in the source image no longer being present in the target image. In SceneNet, transformations between successive camera frames often resulted in occlusions of large areas of the field of view which may have led DeepEfference and LightEfference to incorrectly sample from the target images.

We were surprised by the large performance difference between DeepEfference and LightEfference on the KITTI dataset. As LightEfference is comprised completely of fully-connected layers and neither network uses dropout, one possibility is that LightEfference overfit to the training dataset. This is unlikely as DeepEfference's training Euclidean loss on KITTI was ≈ 50 at $500,000$ training iterations while LightEfference's loss was $\approx 5x$ larger at ≈ 250. This suggests that LightEfference was also performing poorly on the training data and thus most likely not overfitting.

A second possibility is that the large range of depths in scenes in KITTI prevented the limited, 2D-only transformations of LightEfference from learning a single coherent transformation model. This possisibility is supported by the large number of outliers produced by LightEfference which suggests that LightEfference was unable to successfully process the full range of input transforms. For DeepEfference, the mean pixel error percentile scores at 5, 10, 25 and 99 were $percentile(5) = 0.45$, $percentile(25) = 1.07$, $percentile(50) = 1.74$, and $percentile(99) = 11.69$ while for LightEfference, they were $percentile(5) = 0.51$, $percentile(25) = 1.22$, $percentile(50) = 2.21$, and $percentile(99) = 1109.48$. While LightEfference has higher error scores across percentiles, it is the percentile score at 99 that demonstrate its large number of high error outliers which cause its mean pixel error to be an order of magnitude greater than DeepMatching and DeepEfference.

VI. Discussion

We have shown that providing a network with heterogeneous inputs and combining a parametrized global transformation pathway with a pixel-level, local pathway allows for far more computationally efficient predictions with minimal degradation in predictive performance.

Agents must understand which elements in the environment their actions do and not not have the power to affect. A potentially powerful future use for DeepEfference lies in teaching systems what they can and cannot interact with. While deep learning approaches have traditionally been limited in their applications due to their need for large, annotated training sets, DeepEfference's ability to learn without supervision can allow for robots to learn meaningful sensorimotor relationships via bootstrapping simply by operating in an environment.

While the aim of DeepEfference is to generate correspondences between pixels in the source image and pixels in the target image, an unintended side-effect of the predictive training is the generation of image areas where there is no actual overlap between source and target images. Several of these cases are shown in Fig. 4. In first row of Fig. 4, DeepEfference learned to sample from different areas in the target image to imagine what the front of the van looked like despite it not being present in the target image. The same can be seen in the second row where DeepEfference imagines what the left-side of the house looked like.

Performance of the dual-pathway DeepEfference architecture surpassed the global-pathway-only LightEfference architecture in all experimental conditions. As mentioned briefly, we also attempted to use a local-pathway-only architecture but networks trained with this local-only architecture failed to converge. Additional work is needed to determine why these networks failed to learn but we suspect that it is due to the fully-connected layers attempting to learn a complex pixel-level transform beyond their capacity.

Currently, the actual displacement between the camera at the time of each frame are used as transform inputs to DeepEfference. One possible source for this information in real-world robotic applications is from IMUs. However, constant velocity motions may prove difficult for DeepEfference if the expected transforms are being generated from IMU signals. In these cases, it may instead be possible to use a motor command as a surrogate transform signal, but this has yet to be investigated.

Finally, the transform estimates fed to DeepEfference are computed from ground-truth camera poses and do not exhibit noise characteristics that will most likely be found in real-world applications where we will rarely, if ever, have access to a comparably clean extra-visual motion measurement. One possibility for overcoming measurement noise is to expand DeepEfference to produce n image reconstruction predictions and include an additional decision node that chooses the best reconstruction. DeepEfference's runtime could allow for many possible reconstructions to be generated similar to learning and sampling from a noise distribution.

References

[1] 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.

[2] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "SceneNet RGB-D: 5M Photorealistic Images of Synthetic Indoor Trajectories with Ground Truth," *arXiv preprint arXiv:1612.05079*, 2016.

[3] 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.

[4] 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.

[5] T. Kohashi and Y. Oda, "Initiation of Mauthner-or non-Mauthner-mediated fast escape evoked by different modes of sensory input," *The Journal of Neuroscience*, vol. 28, no. 42, pp. 10 641–53, oct 2008.

[6] H. Korn and D. Faber, "The Mauthner cell half a century later: a neurobiological model for decision-making?" *Neuron*, vol. 47, no. 1, pp. 13–28, jul 2005.

[7] J. Brody, D. Perlis, and J. Shamwell, "Who's talking?efference copy and a robot's sense of agency," in *2015 AAAI Fall Symposium Series*, 2015.

[8] J. Brody, S. Barham, Y. Dai, C. Maxey, D. Perlis, D. Sekora, and J. Shamwell, "Reasoning with grounded self-symbols for human-robot interaction," in *2016 AAAI Fall Symposium Series*, 2016.

[9] S. Kumar, F. Odone, N. Noceti, and L. Natale, "Object segmentation using independent motion detection," *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, no. 1, pp. 94–100, 2015.

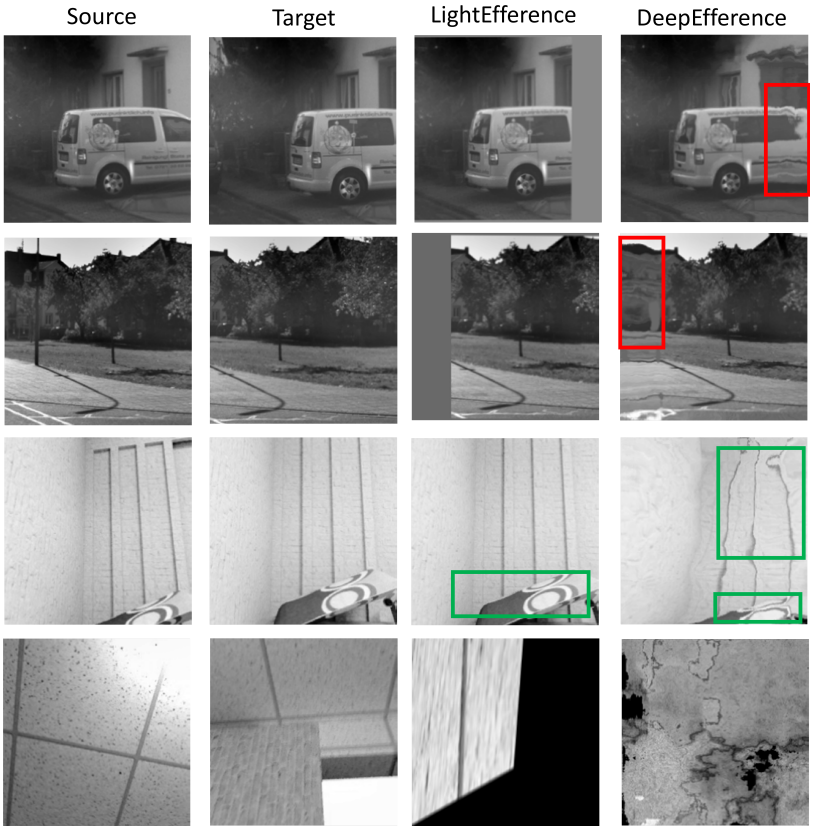

Fig. 4. Unusual DE and LE sample results from KITTI and SceneNet. The red boxes highlight areas in the reconstructed image that were imagined by DeepEfference. Green boxes highlight instances where DE was able to better predict object positions in a scene with strong depth contrast while LE generated a poorer reconstruction. The last row shows a failure case where both LE and DE were unable to generate a reconstruction. This is most likely due to the extreme transformation between the camera at the time of source image capture versus target image capture.

[10] M. Irani and P. Anandan, "A unified approach to moving object detection in 2d and 3d scenes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 6, pp. 577–589, 1998.

[11] R. C. Nelson, "Qualitative detection of motion by a moving observer," *International journal of computer vision*, vol. 7, no. 1, pp. 33–46, 1991.

[12] P. Bhargava, M. T. Cox, T. Oates, U. Oh, M. Paisner, D. Perlis, and J. Shamwell, "The robot baby and massive metacognition: Future vision," in *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1–2.

[13] J. Shamwell, T. Oates, P. Bhargava, M. T. Cox, U. Oh, M. Paisner, and D. Perlis, "The robot baby and massive metacognition: Early steps via growing neural gas," in *Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1–2.

[14] E. Holst and H. Mittelstaedt, "The principle of reafference: Interactions between the central nervous system and the peripheral organs," *PC Dodwell (Ed. and Trans.), Perceptual processing: Stimulus equivalence and pattern recognition*, no. 1950, pp. 41–72, 1950.

[15] R. W. Sperry, "Neural Basis of the Spontaneous Optokinetic Response Produced By Visual Inversion," *Journal of Comparative and Physiological Psychology*, vol. 43, no. 6, pp. 482–489, 1950.

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.

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

[22] 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.

[23] 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.

[24] B. Bridgeman, A. Van der Heijden, and B. M. Velichkovsky, "A theory of visual stability across saccadic eye movements," *Behavioral and Brain Sciences*, vol. 17, no. 2, pp. 247–257, 1994.

[25] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View Synthesis by Appearance Flow," *European Conference on Computer Vision*, vol. 1, pp. 286–301, 2016.

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

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

[28] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[29] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 105–119, 2010.