Reinforcement Learning Based Energy Minimization

Praveen Badimala[1] and Christian Bailer[2]

[1] bgiridha@rhrk.uni-kl.de
[2] christian.bailer@dfki.uni-kl.de

Abstract. In this work, we attempt to predict the optical flow by using the Deep Reinforcement Learning Networks. We use two networks consisting of Actor and Critic network. The training of the network requires no Ground Truth optical flow. We used three different variations of Actor Critic networks having FlowNet Simple [1] as base architecture. We used TVL1 energy function to find the energy of the optical flow from the Actor network and the energy is used as negative reward to train the Actor Critic networks.

Keywords: reinforcement learning, optical flow, tvl1, energy minimization

1 Introduction

Optical flow is the pattern of apparent motion of objects, surfaces and edges in a visual scene caused by the relative motion between observer and the scene [2]. In recent years, the understanding of the visual scene from movable platforms has gained a lot of attention due to the active development of autonomous systems and vehicles. The study on optical flow estimation is being done from the long time. Traditionally, there are many approaches like variational methods, Lucas-Kanade method, but still a holistic solution does not exist. The specularities, object shadow, slow frame rate, reflections and large displacement still pose problem in estimating the accurate optical flow field. The computer vision community has published wide range of algorithms for the estimation of optical flow.

In this project, we find energy of the optical flow using TVL1 energy formulation. The energy is used for the training of the FlowNet Simple network [1] and this is explained in the methodology section 3. The Total Variational L1 energy function is composed of two terms consisting of data fidelity term and regularization term.

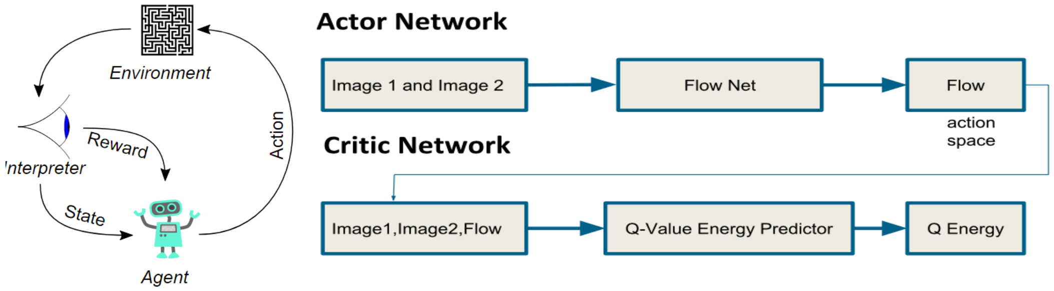
$$E = \int_{\Omega} \Big\{ \lambda |I_0(x) - I_1(x + u(x)| + |\nabla(u)| \Big\} dx \tag{1}$$

The data fidelity term $\lambda |I_0(x) - I_1(x + u(x)|$ computes the image intensity difference at pixel x at image I_0 and the corresponding pixel at image I_1. $u(x)$ denotes the optical flow consisting of $u_1(x)$ and $u_2(x)$ components in x and y direction. The regularization term $|\nabla(u)|$ gives the smoothness of the flow. The TV-L1 implementation by Pérez et al. [3] uses the value 0.15 as default λ value. We use the same λ value for computing the energy of the optical flow generated.

$$E = \int_{\Omega} \Big\{ \lambda |I_0(x) - I_1(x + u(x)| + |\nabla(u_1)| + |\nabla(u_2)| \Big\} dx \tag{2}$$

The energy equation (1) can be rewritten as equation (2) consisting of flow components u_1 and u_2.

Reinforcement Learning is a part of machine learning where an software agent takes an action so that the cumulative reward point increases. Recently many exiting problems involving chess and many games are approached using the techniques of reinforcement learning. Reinforcement Learning algorithm takes an action based on the current state so that it maximizes the future cumulative reward and the reward obtained by taking an action in an environment is taken as a feedback for the reinforcement learning algorithm, the algorithm refines its action based on this reward feedback. Figure 1(a) gives a overview of the Reinforcement learning process. Reinforcement learning acts on an environment represented by state space and takes an action from action space. The recent advances in deep learning has introduced deep networks as action predictors called generally as Deep Q Networks. Deep Q Networks have been used to learn the optimal strategies for playing many atari games [4] and here the actions are predicted by a deep network by taking only the pixel values as the input. These Deep Q networks have achieved human level performance in playing the atari games where the Deep Q networks used for playing games have finite set of actions. The Deep Q networks is also extended to solve problems where action space and state space are continuous and they have shown significant result in control tasks like in the work done by Lillicrap et al. [5].



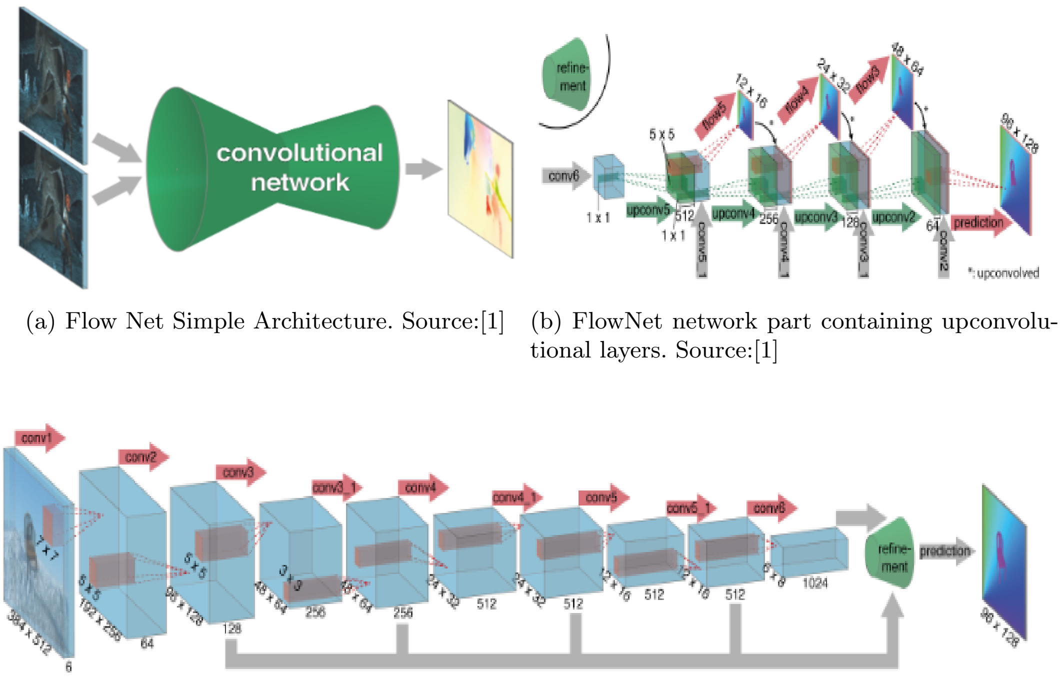
(a) Reinforcement Learning (b) Actor-Critic Network with flow prediction

Fig. 1: Figures describing the RL and RL for optical flow

1.1 FlowNet

We use the FlowNet Simple architecture proposed by Dosovitskiy et al. [1] as the base architecture. FlowNet consists of convolutional, pooling and deconvolutional layers. The image frames I_0 and I_1 are concatenated in a sequence and fed into the FlowNet Simple network as input. the FlowNet Simple network architecture visually appears like that of an auto encoder-decoder network. The method by Dosovitskiy et al. [1] consisted of training the flow network in a supervised manner by taking the ground truth flow. The difference between ground truth flow and the optical flow obtained by the network is considered as loss and the loss is back propagated to the network to train the network.

The Figure 2(a) is the overview architecture of the FlowNet Simple network and the Figure 2(c) explains the convolutional network part. The convolutional network part consists of convolutional layer followed by max pooling layer. The convolutional network part outputs the flow vector in coarse feature maps. The later deconvolutional/upconvolutional network part refines the coarse feature maps to high resolution prediction. The deconvolutional layers have skip connections from the convolutional network part and these skip connections helps in better propagating the gradients from the layers which are near to the flow network output. The refinement or upconvolutional/deconvolutional layers are shown in the Figure 2(b). Dosovitskiy et al. [1] built a dataset to train the Flownet simple network consisting of images obtained from flickr dataset as background and chairs are augmented over the flickr images. The chairs and the background undergo affine transformations to generate the second image and the corresponding flow. We use the flying chairs dataset to train our network. There are 22872 image pairs along with their corresponding ground truth optical flow.



(a) Flow Net Simple Architecture. Source:[1]

(b) FlowNet network part containing upconvolutional layers. Source:[1]

(c) FlowNet network part containing convolutional layers. Source:[1]

Fig. 2: Figures describing the optical flow computation steps

2 Related works

The work done by Yuanlong Li et al. [6] have used Deep Determininstic Policy Gradient(DDPG) algorithm to predict the optimal temperature set points for data center cooling control. The data center power load $\alpha(t)$ and the average ambient temperature T_{amb} inside the room are considered as state variables and the temperature set points for the air conditioners are considered as the action variables. An energy function was formulated based on the data center load and ambient temperature. The energy computed by the energy function is taken as negative reward to train the Actor-Critic Network. They have reduced the energy consumption by 10% compared to the usage of using traditional control algorithms. The Actor-Critic Network is trained on the off-line trace consisting of ambient temperature, power load and temperature set points where as our flow network is trained on on-line trace.

3 Methodology

Our approach is on similar lines of the the work done by Yuanlong Li et al. [6]. We use the energy function as the one which awards the reward/penalty to the network. We use the actor critic networks to predict the optical flow where the energy is given as a feedback to the network. We initially tried the network architecture as shown in the figure 3 and later we further refined the network to better the flow prediction. The algorithm of training the network is explained in the Algorithm section 3.

Algorithm The actor network predicts the optical flow and the critic network predicts the energy of the predicted optical flow. The reference image frame I_0 and next image frame I_1 is given as the inputs to actor network and the actor network predicts the optical flow. The optical flow generated by the actor network is given to the critic network along with image frames I_0 and I_1 to predict their energy. The Figure 1(b) shows the training process. The basic notion is that we alternatively improve the performance of actor network and critic network. The critic networks betters in predicting the energy of the flow and the actor network betters itself in a way so that it minimizes the predicted energy. The below pseudo algorithm gives the description about the training procedure.

Algorithm 1: Training Procedure

```
Data: Training set consisting of $I_0$ and $I_1$ pairs
Result: Trained Actor Network
/* Terms: AN:Actor Network CN:Critic Network pf:predicted flow pq:predicted energy   */
initialize AN and CN with random weights;
for epoch till epoch-max do
    for each $I_0$,$I_1$ batch in epoch-set do
        pf = AN($I_0$,$I_1$);
        pq = CN($I_0$,$I_1$,pf);
        targetEnergy = ComputeTvl1Energy(pf);
        loss = findL1Loss(targetEnergy,pq);
        update CN by minimizing loss;
        pf = AN($I_0$,$I_1$);
        pq = CN($I_0$,$I_1$,pf);
        /* update actor network by the gradient of q network                          */
        update AN by minimizing pq
    end
    store best AN;
    store best CN;
end
```

3.1 AC simple network

The Figure 3 illustrates actor and critic network. The actor and critic network has flownet as base architecture with convolutional, deconvolutional layers and skip connections. The critic network gives energy as output by taking the sum of the last layer of the flow network base.

3.2 AC single network a

There may be a chance that the policy gradients may not be propagated to the intermediate layers of the actor network, so we tried to build the actor and critic network into one single network. We observed a improvement in the flow prediction with single network architecture as discussed in Results section 4. We link the critic network and actor network with skip connections by connecting the convolutional layer in the critic network to the corresponding deconvolution layer in the actor network and this is done by concatenating the upsampling output from the upconvolutional layer of actor network to that of feature maps from convolutional network of critic network. The algorithm remains the same, we train the actor network part and critic network part seperately by updating the weights of the actor network and critic network part alternatively. The skip connections between actor network and critic network helps in better propagation of policy gradients from the critic network to the actor network. Figure 4 shows the network architecture.

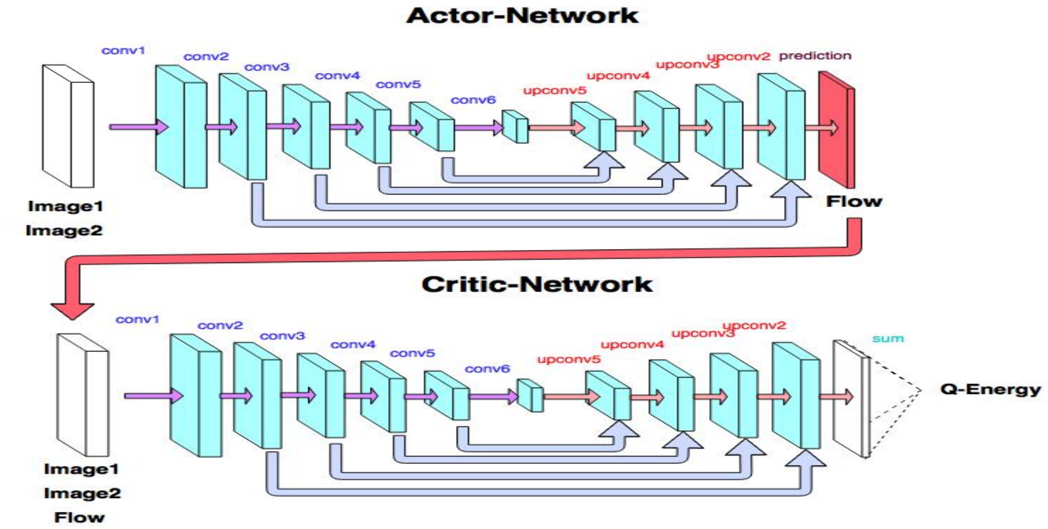


Fig. 3: AC network simple

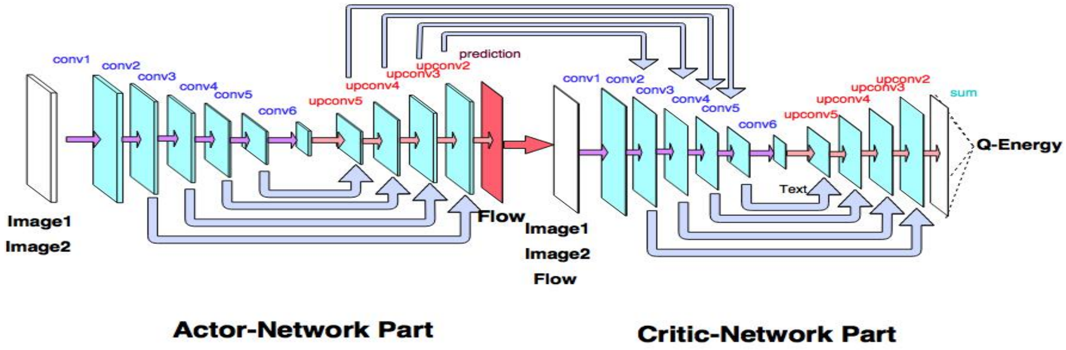


Fig. 4: AC single network a

3.3 AC single network b

In this network architecture, the energy is predicted at each output of the upconvolutional layer of the flow network. The critic network is updated by the sum of the loss computed at each layers. The loss at each layer is computed by taking the difference of energy predicted at that layer in critic network and the corresponding energy of the upsampled flow predicted in the actor network. The results of this network is shown in the Results section 4 and the Figure 5 shows the network architecture.

4 Results

We trained different networks like AC simple, AC single net a and AC single net b with flying chairs dataset and tested with few test samples from the middle bury dataset and also from the flying chairs dataset. AC single net a mpi represents the result of training the network using the mpi-sintel clean dataset [7], rest of all are trained using the flying chairs dataset [1]. The Figure 6 shows the

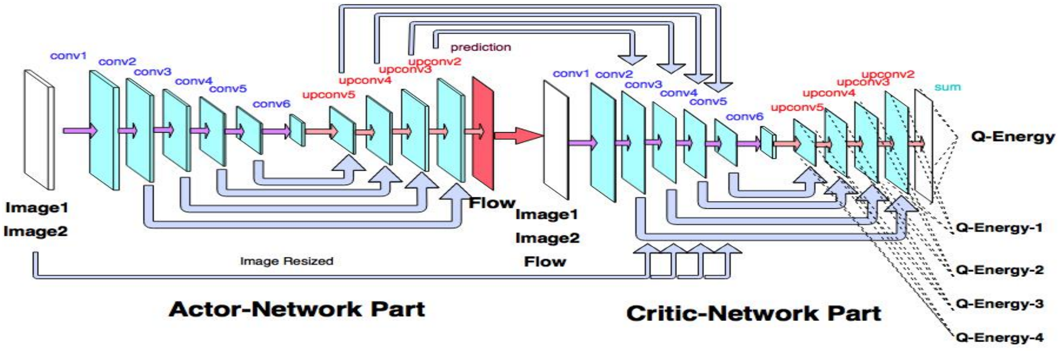


Fig. 5: AC single network b

TVL1 energy of the flow generated by different Actor Critic (AC) networks, Ground truth flow and also the TVL1 algorithm generated optical flow for a set of middle bury test sample images. The Fig 7 shows the TVL1 energy of different flow generated for a set of samples from flying chairs data set. We can observe from the bar graphs 6 and 7 that the AC single net b gives less energy compared to other AC networks. The energy by AC networks are almost double to that of flow generated using TVL1 optimization algorithm.

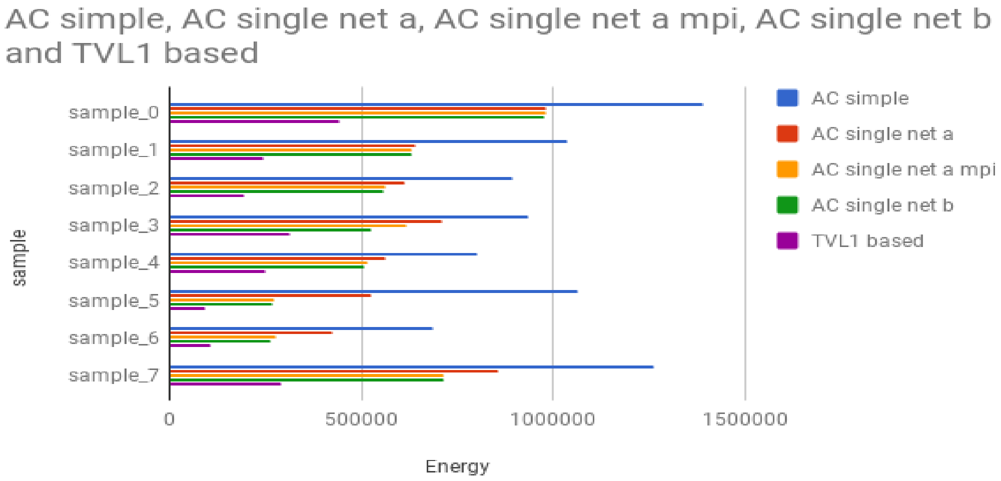


Fig. 6: TVL1 energy of different flows predicted on middle bury test samples

The qualitative results for two sample of flying chairs and middle bury dataset are shown in the Figure 8. The flow maps for different flow are shown and these flow map gives flow variation with respect to a maximum flow value. It is observed from the flow map that the AC network flow shows flow in parts of the image where motion is observed, but the flow is not up to the scale when compared to ground truth flow. We observed that maximum flow values generated by the AC network are in the range of 2 to 5 pixels. The network was not able to predict up-to the scale flow values.

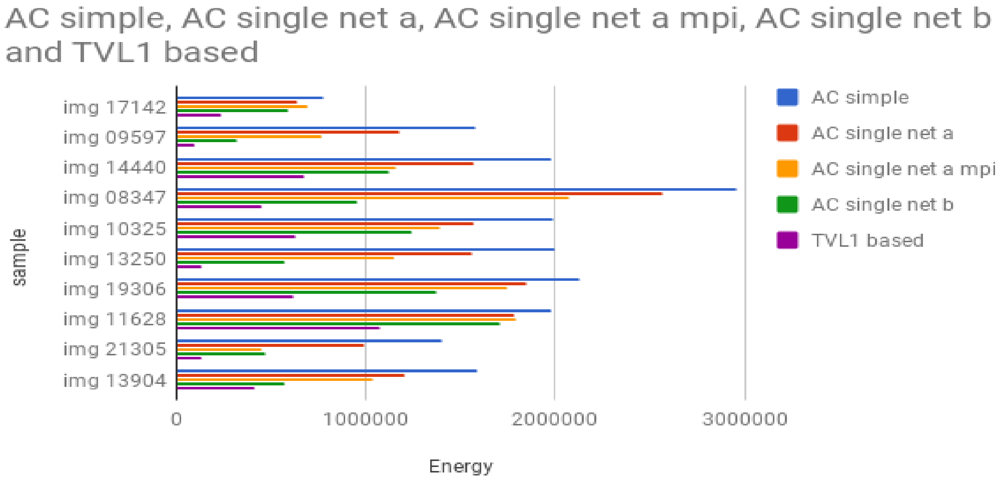


Fig. 7: TVL1 energy of different flows predicted on flying chairs test samples

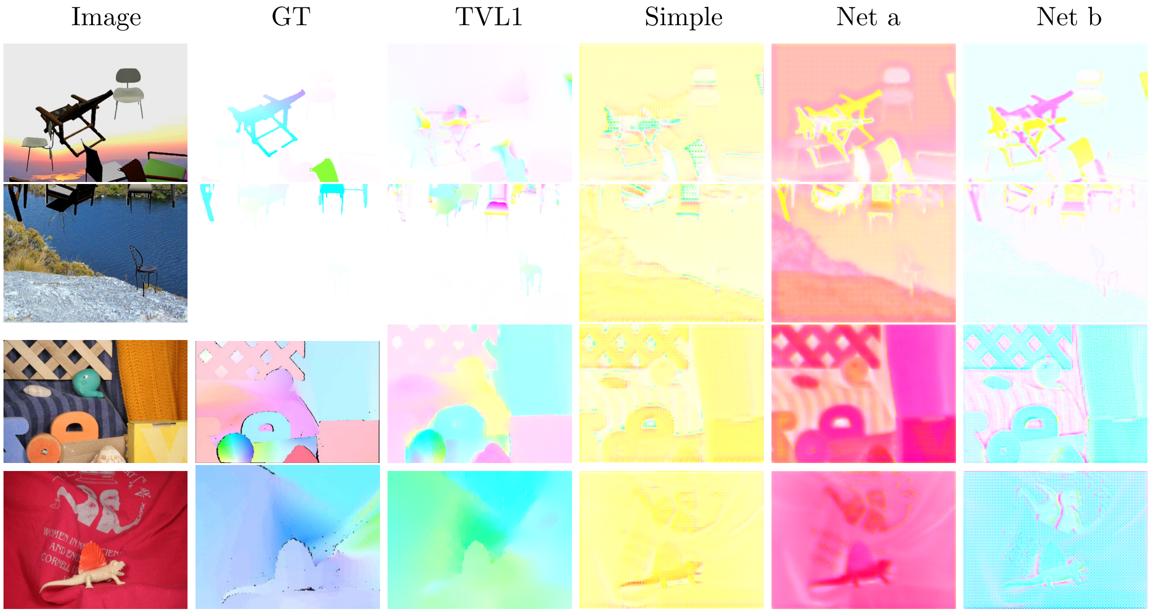


Fig. 8: Qualitative results

5 Conclusion and Further Experiments

We proposed three different actor critic networks and each one gave better result incrementally with increasing the complexity of the network. We have tried a new approach for training optical flow network using the reinforcement learning approach. Although the results of the optical flow generated from the Actor Critic networks were not comparable to the optical flow found using TVL1 optical flow and the energy is double to that of TVL1 optical flow, we trained the network in a unsupervised way. The flow generated from the actor network shows flow in the region of image parts where motion is occurring. With this approach, we have also showed that there is a possibility of training a neural network with just an energy function. As further experiments, one can use a fully connected layer in the critic network at the last layer before the energy prediction layer. One can also increase the depth of the actor and critic networks and use FlowNet2 [8] with correlation layer instead of FlowNet simple network. We used λ value of 0.15 in finding of the energy, instead one can use different λ values and train the network and also scale the smoothness term. We used TVL1 energy function, one can also use different energy functions like Horn and Schunk energy function [9]. As future research, we can try to use an energy function of any domain to train a neural network and check how good the neural network performs for that domain problem and one can suggest to use the neural network as an alternative to the optimization algorithm.

References

1. Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.
2. Andrew Burton and John Radford. *Thinking in perspective: critical essays in the study of thought processes*. Methuen, 1978.
3. Javier Sánchez Pérez, Enric Meinhardt-Llopis, and Gabriele Facciolo. Tv-l1 optical flow estimation. *Image Processing On Line*, 2013:137–150, 2013.
4. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
5. Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
6. Yuanlong Li, Yonggang Wen, Kyle Guan, and Dacheng Tao. Transforming cooling optimization for green data center via deep reinforcement learning. *arXiv preprint arXiv:1709.05077*, 2017.
7. Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625. Springer, 2012.
8. Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
9. Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.