# Unsupervised learning of depth and motion

Kishore Konda
Goethe University Frankfurt
Germany
konda@informatik.uni-frankfurt.de

Roland Memisevic
University of Montreal
Canada
roland.memisevic@umontreal.ca

## Abstract

*We present a model for the joint estimation of disparity and motion. The model is based on learning about the interrelations between images from multiple cameras, multiple frames in a video, or the combination of both. We show that learning depth and motion cues, as well as their combinations, from data is possible within a single type of architecture and a single type of learning algorithm, by using biologically inspired "complex cell" like units, which encode correlations between the pixels across image pairs. Our experimental results show that the learning of depth and motion makes it possible to achieve state-of-the-art performance in 3-D activity analysis, and to outperform existing hand-engineered 3-D motion features by a very large margin.*

## 1. Introduction

A common property of 3-D inference and motion estimation is that both rely on establishing correspondences between pixels in two (or more) images. For depth estimation, these are correspondences between multiple views of a scene, for motion estimation between multiple frames in a video. Despite superficial differences between these tasks, such as the typical size of the average displacement across images, or whether the geometry is constant or variable across pairs, there are much stronger commonalities between the tasks, such as the fact that both rely on finding positions in one image which match those in another image. This suggests that both tasks may be learnable using essentially the same type of architecture and the same type of learning algorithm, but there has been hardly any work on trying to exploit this in practice. Besides the obvious advantage of allowing us to develop and maintain a single piece of code to achieve both tasks, it makes it trivial to fuse the information from both sources and thereby to design architectures that learn representations of multi-camera video streams with application, for example, in activity analysis.

In the neuroscience literature, the so-called complex cell "energy model", is assumed to be the main underlying mechanism behind both depth and motion estimation (e.g. [1, 3]), and it provides an elegant explanation for how the brain can learn both using the same type of neural hardware.

There has been some progress recently in learning motion energy models from data [10, 17], and learning based methods are among the state-of-the-art in activity analysis from videos. However, there has been hardly any work on learning energy models for depth inference, nor for learning depth and motion information at the same time. In this work we show that it is, in fact, possible to *learn* about 3-D depth entirely unsupervised from data, similar to learning motion as done using complex cell type models. Our experiments show how this makes it possible to achieve state-of-the-art performance in 3-D activity analysis from multi-camera video without making use of any hand-crafted features.

### 1.1. Biologically inspired models of correspondence

The first step to infer depth from two views is to find correspondences between points which represent the same 3-D location [7]. The two standard ways to approach this task are: 1.) For each position in one image find a nearby matching point in the other image using some measure of similarity between local image patches (e.g. [16]). 2.) For each position in both images, extract features that describe phase and frequency content of the region around that point, and read off the phase difference across the two images from the set of filter responses [14].

The first approach has been more common in practice, although the second is more biologically plausible, as it does not require loops over local patches. More importantly, the second approach is amenable to *data-driven learning* as we shall show. The most well-known account of phase-based disparity estimation is the binocular energy or cross-correlation model (e.g. [14, 4, 13]). In its most basic form, this model states that local disparities are encoded in the sum of the squared responses of two neurons, each of which has a binocular receptive field. Each binocular receptive field, in turn, shows a position-shift across the two views.

Between them the two receptive fields show a quadrature relationship (within each view). It can be shown that the position-shift across the views allows the energy model to encode local disparity, while the quadrature relationship within each view allows it to be independent of the Fourier phase of the local stimulus [14, 3].

Analogous models, also based on energy or cross-correlation, have been proposed independently for motion encoding [3, 2, 12]. This is not surprising if one considers that motion can be defined as the transformation of a given input over time and disparity as the transformation of the input across multiple views or a stereo pair. If the given input is a set of frames from a time sequence the model encodes motion and when the input is a stereo pair it encodes disparity. It has been proposed that, due to similarity of models for motion and disparity encoding, it should be possible to integrate them [14]. But to date, there has been no practical exploration of this idea, nor of the *learning* of depth from data.

In this paper we present an approach to learning depth, motion and their combination from data, by using a feature learning architecture based on the energy model. Our approach is based on the view of energy model proposed by [9], which shows that the (motion) energy model can be viewed as two independent contributions to motion encoding: 1) the detection of spatio-temporal "synchrony", and 2) the encoding of invariance. [9] present an autoencoder model using multiplicative interactions for detection of synchrony, and they show that a pooling layer independently trained on the hidden responses can be used to achieve content invariance. We adopt that approach for the estimation of depth and motion cues, as it gives rise to an efficient single-layer learning algorithm. But there a variety of learning based energy models that one could use instead (e.g., [10, 17]).

A description of the synchrony condition and how it can be used for implicit encoding of depth is presented in the next section. Since depth is encoded implicitly in the feature responses of the model, we then show how it is possible to "calibrate" an energy model learned on stereo data using available ground truth data to compute an explicit depth map from this encoding. Since in most applications, the representation of depth is a means to an end not a goal on its own, we then explore a variety of ways to utilize the implicit encoding of depth, as well as motion, using the same approach to learning features. We evaluate and compare several variations of this approach on the Hollywood3D activity recognition dataset [6], and we demonstrate that it improves significantly upon the state-of-the-art, using a minimum of hand engineering.

## 2. Depth as a latent variable

The classic energy model (e.g. [1, 3]) states that we can obtain an estimate of the transformation, $P$, between two images $\vec{x}_1$ and $\vec{x}_2$ by computing a weighted sum over products of filter responses on the images. In particular, if the filters themselves differ by the transformation $P$, so that,

$$\vec{w}_2 = P\vec{w}_1 \tag{1}$$

then the product filter responses will be large for input images for which $\vec{x}_2 = P\vec{x}_1$ holds, too. This makes it possible to extract motion, if $\vec{x}_1$ and $P\vec{x}_2$ denote adjacent frames in a video, and disparity if they denote two patches cropped from the same position of a stereo pair. In most practical situations (for both motion and disparity estimation), the dominant transformation between the images is a *local translation*, in which case the optimal filters are Gabor features and $P$ is a small phase shift. In early, biologically motivated approaches to estimating displacements, filters have been hand-coded [1, 3]. In the context of motion estimation, various approaches were proposed recently to learning the filters from data (e.g. [17, 10, 11]). While learning has been inefficient due to the vast amounts of image patch pairs required for learning good filters, [9] recently presented the "synchrony autoencoder", which learns motion representations more efficiently, using a single-layer autoencoder with multiplicative interactions. We use a similar approach for defining models that learn to encode depth. We shall review that model, as well as show how we can use it for depth and motion estimation in the following section.

### 2.1. Depth across stereo image pairs

Based on the above description, we can define a model based on the synchrony autoencoder (SAE) [9] for learning depth representation from stereo pair of images as follows. Assume we are given a set of stereo image pairs, $\vec{x}, \vec{y} \in \mathbb{R}^N$. Let $\mathbf{W}^x, \mathbf{W}^y \in \mathbb{R}^{Q\times N}$ denote the matrices containing $Q$ feature vectors $\vec{W}_q^x, \vec{W}_q^y \in \mathbb{R}^N$, stacked row-wise.

We define the latent representations $\vec{f}^x = \mathbf{W}^x\vec{x}$ and $\vec{f}^y = \mathbf{W}^y\vec{y}$, which are typically called *factors* in the context of energy models [17, 11, 9]. The hidden representation of disparity is then defined as

$$\vec{h} = \sigma(\vec{f}^x \odot \vec{f}^y) \tag{2}$$

where $\sigma = (1 + \exp(-x))^{-1}$ is a saturating non-linearity. (We use the logistic sigmoid in this work, but other non-linearities could be used as well.)

A standard way to train an autoencoder is by minimizing reconstruction error. Since the vector of multiplicative interactions between factors represents the transformation between $x$ and $y$, here we may define the reconstruction of

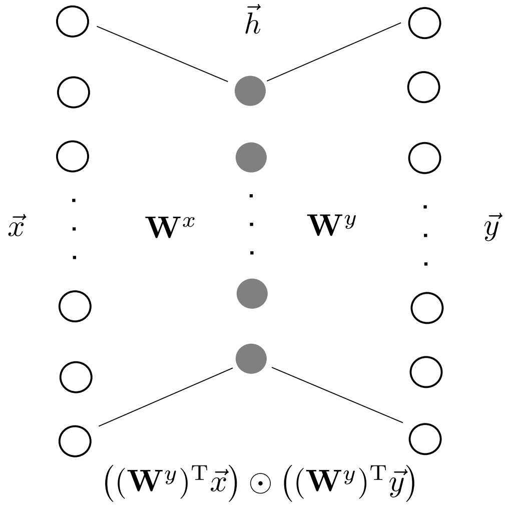

$\vec{h}$
$\vec{x}$
$\mathbf{W}^x$
$\mathbf{W}^y$
$\vec{y}$
$\big((\mathbf{W}^y)^{\mathrm{T}}\vec{x}\big) \odot \big((\mathbf{W}^y)^{\mathrm{T}}\vec{y}\big)$

Figure 1: A model encoding the transformation inherent in the image pair $(\vec{x}, \vec{y})$.

one input given the other input *and* the transformation as

$$\hat{x} = (\mathbf{W}^x)^{\mathrm{T}}(\vec{h} \odot \vec{f}^y) \tag{3}$$

$$\hat{y} = (\mathbf{W}^y)^{\mathrm{T}}(\vec{h} \odot \vec{f}^x) \tag{4}$$

Here, we assume an autoencoder with tied weights similar to [17, 11, 9]. This allows us to define the reconstruction error the as symmetric squared difference between inputs and their corresponding reconstructions:

$$L((\vec{x}, \vec{y}), (\hat{\vec{x}}, \hat{\vec{y}})) = \|(\vec{x} - \hat{\vec{x}})\|^2 + \|(\vec{y} - \hat{\vec{y}})\|^2 \tag{5}$$

### 2.1.1 Regularization

For extraction of sparse and robust representation we use contraction as regularization [15] which amounts to adding the Frobenius norm of the Jacobian of the hiddens with respect to the inputs $x, y$.

$$\|J_e(\vec{x}, \vec{y})\|_E^2 = \sum_{ij} \left( \frac{\partial h_j(\vec{x}, \vec{y})}{\partial x_i} \right)^2 + \sum_{ij} \left( \frac{\partial h_j(\vec{x}, \vec{y})}{\partial y_i} \right)^2 \tag{6}$$

Using sigmoid non-linearity the contraction term becomes

$$\begin{aligned} \|J_e(\vec{x}, \vec{y})\|_E^2 = & \sum_j (h_j(1-h_j))^2 (f_j^x)^2 \sum_i (W_{ij}^x)^2 \\ & + \sum_j (h_j(1-h_j))^2 (f_j^y)^2 \sum_i (W_{ij}^y)^2 \end{aligned} \tag{7}$$

Thus the complete objective function employing contractive regularization, using $\lambda$ as the regularization strength, is

$$\mathcal{J}_C = L((x, y), (\hat{x}, \hat{y})) + \lambda \|J_e(x, y)\|_E^2 \tag{8}$$

To obtain filters that represent depth we minimize Eq. 8 for a set of image pairs cropped from identical positions of multiple views of the same scene. It is important to use a patchsize that is large enough to cover the maximal disparity in the data, otherwise the model will not be able to encode the corresponding depth. In contrast to traditional approaches to estimating depth, however, there is no need for rectification, since the model can learn any transformation between the frames not just horizontal shift.

## 2.2. Depth across stereo sequences

In the previous section we described a model for encoding depth across stereo image pairs. We now propose several extensions of this approach to learn representations from stereo sequences not still images. This makes it possible to extract a representation informed by both motion and depth from the sequence. We defer the detailed quantitative evaluation of the approaches to Section 3.

### 2.2.1 Encoding depth

Let $\vec{X}, \vec{Y} \in \mathbb{R}^N$ be the concatenation of $T$ vectorized frames $\vec{x}_t, \vec{y}_t \in \mathbb{R}^M, t = 1, \dots, T$, and be defined such that $(\vec{x}_t, \vec{y}_t)$ are stereo image pairs. Let $\mathbf{W}^x, \mathbf{W}^y \in \mathbb{R}^{Q \times N}$ denote matrices containing $Q$ feature vector pairs $\vec{W}_q^x, \vec{W}_q^y \in \mathbb{R}^N$ stacked row-wise. Each feature is composed of individual frame features $\vec{w}_{qt}^x \in \mathbb{R}^M$ each of which spans one frame $\vec{x}_t$ from the input sequence. Accordingly for the features in $\mathbf{W}^y$.

In analogy to the previous section, we can define the factors $\vec{F}^X = \mathbf{W}^x \vec{X}$ and $\vec{F}^Y = \mathbf{W}^y \vec{Y}$ corresponding to the sequences $\vec{X}, \vec{Y}$. A simple representation of depth may then be defined as

$$H_q^D = \sigma(F_q^x \cdot F_q^y) \tag{9}$$

The representation $H_q^D$ will contain products of frame responses $\big((\vec{w}_{qt}^x)^T \vec{x}_t\big) \cdot \big((\vec{w}_{qt}^y)^T \vec{y}_t\big)$ which detect synchrony over stereo pairs encoding depth. It will also contain products across time and position, $\big((\vec{w}_{qt}^x)^T \vec{x}_t\big) \cdot \big((\vec{w}_{q(t+i)}^y)^T \vec{y}_{t+i}\big)$, which will weakly encode motion as well. In other words, motion is encoded indirectly by this model, by computing products of responses at different times across cameras. We shall refer to this model as **SAE-D** for "depth encoding synchrony autoencoder" in the following.

### 2.2.2 Encoding motion

For analyzing the effect of encoding motion vs. depth on the classification of sequences, we can define a hidden representation which employs only a single stereo sequence as follows. Let $\vec{X}$=$\vec{Y}$ represent a single camera channel from the stereo sequence. If we tie the weight matrices $\mathbf{W}^x, \mathbf{W}^y$

to be identical as well, Eq. 9 may be rewritten

$$H_q^M = \sigma((F_q^x)^2) \tag{10}$$

Since $F_q^x$ is the sum over individual frame filter responses, its square, by the binomial identity, will contain products of individual frame responses across time $\big((\vec{w}_{qt}^x)^T\vec{x}_t\big) \cdot \big((\vec{w}_{q(t+i)}^x)^T\vec{x}_{t+i}\big)$ as well as the squares of filter responses on individual frames. $H_q^M$ will therefore take on a large value only for those filters which match all indvidual frames, which implies that they will jointly satisfy Eq. 1. This observation is the basis for the well-known equivalence between the energy model and the cross-correlation model (see, for example, [3, 12, 9]).

Thus, in this case synchrony is detected over time, encoding the motion present in the input sequence. In analogy to the previous section, the encoding of motion will be weakly related to depth in the scene, as well, because depth and motion tend to be correlated. Any camera motion, for example, may be viewed as providing multiple views of a single scene, thereby implicitly containing information about depth (a fact that is exploited in structure-from-motion approaches). However, due to the absence of camera motion which is *consistent* across the dataset, as well as the presence of a multitude of object motions, the depth information will only be weakly present in any encoding of motion. We shall call this model for representing motion **SAE-M** in the following. The model is equivalent to the SAE defined in [9] for encoding motion from a single-channel video.

### 2.2.3 Multiview disparity

To obtain an explicit encoding of both depth and motion, we require the detection of synchrony both across time and across stereo-pairs. One way to obtain such a representation in practice is to combine the representations defined in the previous two sections, for example, by using their average or concatenation.

As a third alternative, we propose defining a joint representation by including products of frame responses across both time and stereo-pairs. Recall that the square of the sum over frame-wise filter responses contains within-channel motion information. We suggest obtaining an estimate of the across-channel disparity information by defining the hidden unit response as the product over theses squares. This allows us to extract information about disparity from the relation between the temporal evolutions of the complete video sequence, rather than between feature positions across single frames. To this end, we define the hidden representation

$$H_q^{MD} = \sigma((F_q^x)^2 \cdot (F_q^y)^2) \tag{11}$$

The representation $H_q^{MD}$ may be written as $\big(\sum(\vec{w}_{qt}^x)^T\vec{x}_t\big)^2 \cdot \big(\sum(\vec{w}_{qt}^y)^T\vec{y}_t\big)^2$, and it may be thought of as a “multi-view” or “motion-based” estimate of disparity. We call a model based on this representation of disparity **SAE-MD** in the following.

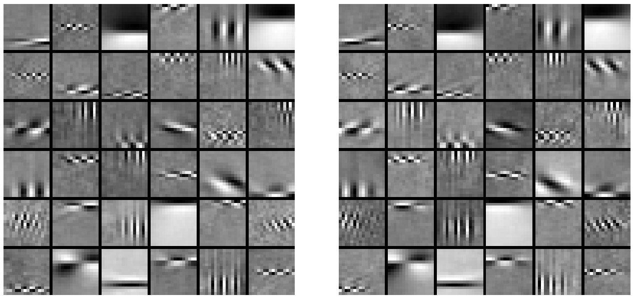


Figure 2: Filters learned on stereo patch pairs from the KITTI dataset.

## 2.3. Learning

For the models described above the decoder and reconstruction cost can be derived to be similar to that of stereo-pair model in Section 2.1. In particular, the reconstruction error and contraction cost for the models SAE-D and SAE-M can be derived by replacing the corresponding parameters of Equations 5 and 7. For the SAE-D model this amounts to replacing frames $\vec{x}, \vec{y}$ with sequences $\vec{X}, \vec{Y}$, and for the SAE-M model to further substituting $\vec{X}$ for $\vec{Y}$.

For the SAE-MD model, we found the contraction cost to be unstable due to presence of higher exponents in the hidden representation. Because of this, we use the trained weights from the SAE-D model and during inference use the representation from Equation 11. Alternatively it may be possible to train the model using a denoising criterion instead of contraction for regularization.

## 2.4. Interest point detector

Hand-crafted image, motion or 4-D descriptors are typically accompanied by corresponding interest point detectors. Since they reduce the number of positions to extract representations from, they have been shown to improve efficiency and performance, for example, in bag-of-features based recognition pipelines.

For a learned representation that is based on the linear projection of image patches, it is possible to define a default interest point operator, by using norm-thresholding of feature activations (see, for example, [10]). It can be motivated by the observation that norms of relevant features will be higher at edge and motion locations than at homogeneous or static locations [8]. Norm thresholding interest point detection amounts to simply discarding features $H$ with norm $|H|_1 < \delta$. The value of $\delta$ may be chosen based on the mean norm of the features in the training set.

# 3. Experiments

## 3.1. Learning depth from image pairs

It has been well-known that energy and cross-correlation models with hand-crafted Gabor features are able to extract depth information from random-dot stereograms (e.g., [14]). In order to test whether depth information can be extracted in more realistic settings and using features that are learned from data, as proposed in Section 2, we first conducted an experiment where a depth map is estimated given a stereo image pair. For this experiment we use stereo images from the KITTI stereo/flow benchmark [5]. The dataset consists of 194 training image pairs and 195 test image pairs. For the training image pairs corresponding ground truth depth is provided. Since the ground truth is captured by means of a Velodyne sensor which is calibrated with the stereo pair it is only provided for approximately 30% of their image pixels. We down-sampled the images from a resolution of $1226 \times 370$ pixels to $300 \times 100$ pixels, so that the local shift between image pairs falls within the local patch size, which is a crucial requirement for models using local phase matching for disparity computation as discussed in Section 2.

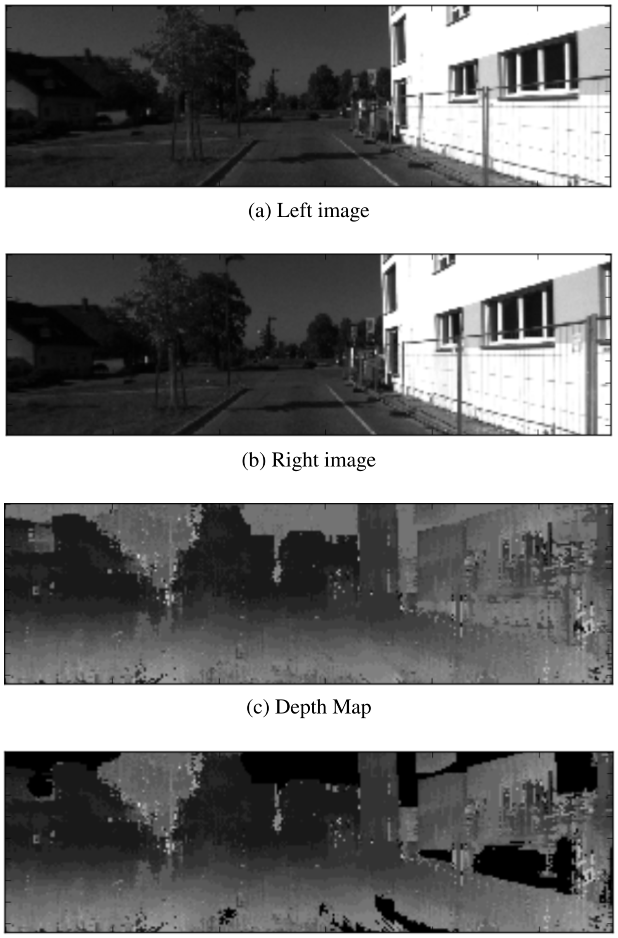
We trained the stereo-pair model described in Section 2.1 (Eq. 2) on patch pairs cropped from the training set. Each patch is of size $16 \times 16$ pixels and the total number of training samples is $100,000$. The patches used for learning the filters are cropped only from regions of images where corresponding depth information is available.

Some learned filters are shown in Figure 2. The figure shows that filters are localized, Gabor-like and span a wide range of frequencies and positions. Since cameras are parallel the filters learned predominantly horizontal shifts.

To test if we can extract depth information from the learned hidden representation, we trained a logistic regression classifier using the available ground truth as the output data. To this end, we generate labels by taking the mean over non-zero pixel intensities of corresponding patches from the ground truth, which we then quantize into 25 bins. After training the classifier, estimation of depth for a given stereo pair involves dense sampling of patch pairs followed by feature computation and prediction by the classifier.

A sample stereo image pair and the learned depth map is shown in Figure 3. In the figure, each predicted depth label is one pixel of the estimated depth map. An artifact of this depth estimation procedure is that object boundaries are expanded over their actual size due to the patch size used in the model. It can be also observed that the depth for featureless regions like sky and plane surfaces is less accurate than in feature rich regions, because the model cannot detect any shift in those cases.

This is true, of course, for any disparity estimation scheme based on local region information, and when the goal is an explicit depth map, one should use a Markov Random Field or similar approach to cleaning up the obtained depth map. In the event where one is not interested in an exact depth map, but rather in depth cues to help make predictions that merely *depend* on depth (similar to the bag-of-features approach taken typically in motion estimation), a possible alternative is the use of an interest point detector as explained in Section 2.4. Figure 3c shows an example of an estimated depth map with interest points, and it shows that norm thresholding masks out most of the regions predominantly homogeneous regions in the image.


(a) Left image

(b) Right image

(c) Depth Map

(d) Depth map masked using interest points

Figure 3: Stereo image pair and estimated depth maps. Depth map scale ranges from 1 (Far, shown as black) to 25 (Near, shown as white).

In general, we thus observe that it is possible to infer depth information from the filter responses defined in Section 2, even if the information comes in the form of noisy cues, similar to most common estimates of motion, rather than in the form of a clean depth map. We shall discuss an approach to exploiting this information in a bag-of-features

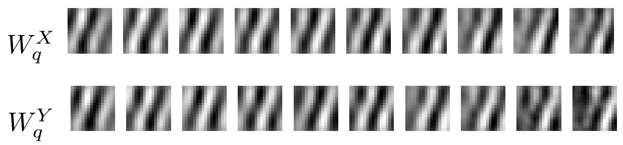


Figure 4: Example of a filter pair learned on sequences by the SAE-D model from the Hollywood3D dataset.

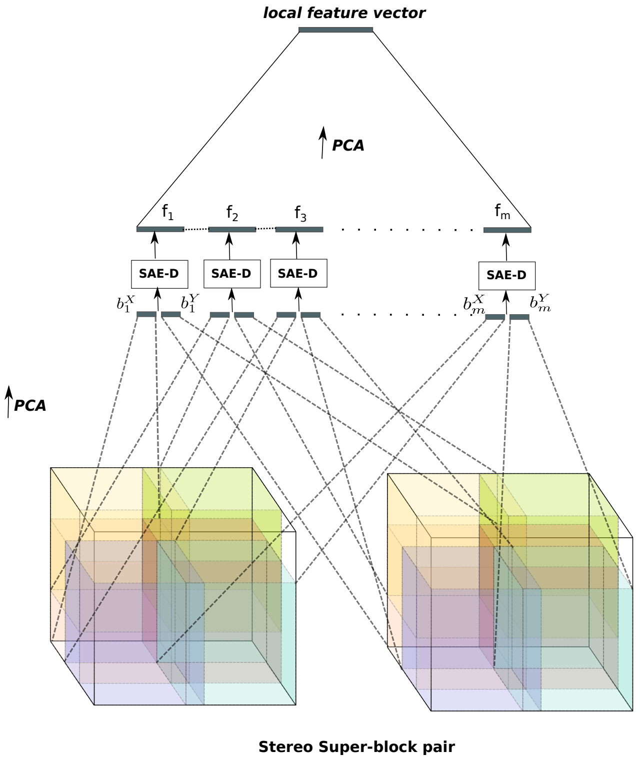


Figure 5: Feature extraction framework using the SAE-D model.

pipeline for activity recognition in the next section.

## 3.2. Activity Recognition

We evaluate the effect of implicit depth encoding on the task of activity recognition, using the Hollywood3D dataset introduced by [6]. The dataset consists of stereo video sequences along with computed depth videos. The videos are of 14 different categories with 643 videos for training and 302 for testing. The different categories are 'Run', 'Punch', 'Kick', 'Shoot', 'Eat', 'Drive', 'UsePhone', 'Kiss', 'Hug', 'StandUp', 'SitDown', 'Swim', 'Dance' and 'NoAction'. The videos are downsampled spatially from size of $1920 \times 1080$ to $320 \times 240$. Models are trained on PCA whitened spatio-temporal block pairs with each block of size $10 \times 16 \times 16$. $300,000$ samples are used for training and the number of hidden units is fixed for all models to 300. A sample feature pair learned by the SAE-D model is shown in Figure 4. Each filter in the pair spans ten frames. The filters are again Gabor-like and show a continuous phase shift through time, and another phase shift across camera views.

For the quantitative evaluation, we use the framework presented by [6]. After performing feature extraction, we perform K-means vector quantization followed by a multi-class SVM with RBF kernel for classification. A flow diagram of the pipeline is visualized in Figure 6.

Feature are extracted using a convolutional architecture similar to that presented in [9, 10]. Super block pairs of size $14 \times 20 \times 20$ pixels each are cropped densely with stride $(7, 10)$ in time and space, respectively, from the stereo video pairs. From the super blocks, sub-blocks of the same size as the training block size ($10 \times 16 \times 16$ pixels) are cropped with stride $(4, 4)$, resulting in 8 sub-blocks per super block.

We first compute the feature vector for each stereo sub-block pair. We then concatenate feature vectors corresponding to the sub-blocks of a super block and reduce their dimensionality using PCA. This procedure, using the SAE-D model as an example, is visualized in Figure 5. The number of words for K-means vector quantization is set to 3000.

Our main goal in these experiments is to evaluate the impact of the implicit depth encoding in the task of activity recognition. We compare a variety of settings to this end. In **experiment 1**, the SAE-D is used for feature extraction. As we discussed in Section 2 the SAE-D primarily encodes depth. **Experiment 2** uses the SAE-M for feature extraction with only one of the stereo channels as input. **Experiment 3** employs the SAE-MD for features extraction, and is thus based on a representation that integrates across-frame and across-channel correlations. In **Experiment 4** we test two alternative ways of integrating depth and motion information, by combining the representations from two separately trained SAE-D and SAE-M models. The first, which we call SAE-MD(Ct), amounts to concatenating the representations from SAE-D and SAE-M as features. The other, SAE-MD(Av), amounts to computing the average precision using the mean over confidences from experiments 1 and 2. Thus it amounts to averaging the classification decisions of two separate classification pipelines (one based primarily on depth, and the other based primarily on motion).

Each configuration is evaluated by computing the average precision and the correct-classification rates. The results are reported in Tables 1 and 2. We repeated the experiments using the norm-thresholding interest point detector described in Section 2.4.

From the results it can be observed that the combination of motion and depth cues performs better than using individual cues. All results, including the motion-only, the depth-only and the combination models, outperform all existing models, based on hand-crafted representations, by a

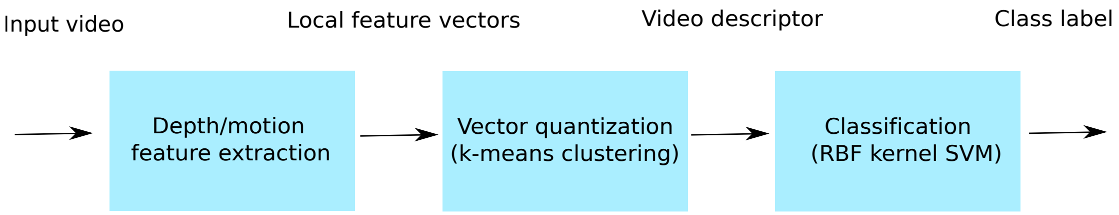

Input video
Local feature vectors
Video descriptor
Class label
Depth/motion feature extraction
Vector quantization (k-means clustering)
Classification (RBF kernel SVM)

Figure 6: Flow diagram of the classification pipeline.

| **Method** | **Interest points** | **AP** | **CC Rate** |
|---|---|---|---|
| SAE-M | None | 24.31 | 29.61 |
| SAE-MD | None | 24.69 | 29.47 |
| SAE-D | None | 23.53 | 26.82 |
| SAE-M | N-Th | 24.61 | 31.79 |
| SAE-MD | N-Th | 25.14 | 30.46 |
| SAE-D | N-Th | 24.05 | 26.49 |
| SAE-MD(Ct) | N-Th | 24.45 | 29.47 |
| SAE-MD(Av) | N-Th | 26.11 | 30.13 |
| HoG/Hof/HoDG [6] | 3.5D-Ha | 14.1 | 21.8 |
| RMD-4D[6] | 3D-Ha | 15.0 | 15.9 |

Table 2: Correct classification rate and average precision.

very large margin, and they are to the best of our knowledge the best reported results on this task to date.

It has been observed in the past that learning based features tend to outperform more traditional features, like SIFT, in object recognition tasks and, more recently and by a larger margin, in motion analysis tasks as compared to spatio-temporal variations of SIFT (e.g., [10, 17, 9]). This observation is confirmed in this 4-D dataset, where it seems to be even more pronounced.

We can also observe that models using interest points (cf., Section 2.4) provide an additional consistent (albeit smaller) improvement over those that do not. Furthermore, the use of depth information provides an edge over motion-only models. Interestingly, the overall effect of the various variations of the model differ heavily across action class, which can be seen in Table 1. For example, the AP for classes Run, Kick, Shoot and Eat are the highest when using primarily depth features for classification; NoAction and Kiss have best AP when using just motion features; and the AP for all the other classes is the highest when combining depth and motion features. This can be due to multiple reasons, and it is likely related to the average depth variation within the activity class. A detailed analysis of which type of information is the most useful for which type of activity class is an interesting direction for future work.

A well-known and popular "recipe" to improve performance in learning tasks has been to base classification decisions on the combination of multiple different models, each of which utilizes a different type of feature. While this recipe often works well in practice, the main challenge to make it work is to develop models which are sufficiently *different* from one another, so they yield a sufficiently large reduction of variance. Utilizing the combination of depth and motion cues may be viewed in this context also as a way to extract cues from video data which are different, since they represent very different properties of the environment.

## 4. Discussion

Most current practical work on stereopsis focuses on extracting dense depth-maps using MRFs. Potential reasons for biology to take a different route might be that (a) depth via deep learning makes it possible to use the exact same learning algorithm for depth inference that is also used to recognize objects and motion; (b) a simple depth cue, as given by a feature vector, $H$, is often entirely sufficient to take swift vital decisions, such as to dodge an approaching object; (c) learning depth inference from data allows for feed-forward depth perception, and thus to avoid the need for a complicated and brittle pipeline, which involves rectification, hypothesis generation, and robustification using RANSAC [7].

In this paper we showed how unsupervised feature learning may be used to mimic this way of extracting depth cues from image pairs, and that learning joint representations of motion and depth within a single type of architecture and a single type of learning rule can achieve state-of-the-art performance in a 3-D activity recognition task. Our work is to the best of our knowledge the first published work that shows that deep learning approaches, which have hitherto been shown to work well in object and motion recognition tasks, are also applicable in the domain of depth inference, or more generally to 3-D vision.

| Action | SAE-MD | SAE-MD(Av) | SAE-MD(Ct) | SAE-M | SAE-D | 3D-Ha | 4D-Ha | 3.5D-Ha |
|---|---|---|---|---|---|---|---|---|
| *NoAction* | 12.10 | 12.77 | 13.10 | **15.73** | 12.15 | 12.1 | 12.9 | 13.7 |
| *Run* | 52.56 | 50.44 | 51.45 | 45.38 | **56.07** | 19.0 | 22.4 | 27.0 |
| *Punch* | **41.09** | 38.01 | 32.68 | 33.86 | 36.17 | 10.4 | 4.8 | 5.7 |
| *Kick* | 9.41 | 7.94 | 6.86 | 6.63 | **11.84** | 9.3 | 4.3 | 4.8 |
| *Shoot* | 30.26 | 35.51 | 30.49 | 30.52 | **40.72** | 27.9 | 17.2 | 16.6 |
| *Eat* | 5.85 | 7.03 | 6.78 | 7.29 | **9.03** | 5.0 | 5.3 | 5.6 |
| *Drive* | 52.65 | 59.62 | 51.35 | 61.61 | 45.19 | 24.8 | 69.3 | **69.6** |
| *UsePhone* | 22.79 | **23.92** | 19.01 | 23.60 | 23.36 | 6.8 | 8.0 | 7.6 |
| *Kiss* | 15.03 | 16.40 | 16.12 | **17.86** | 17.06 | 8.4 | 10.0 | 10.2 |
| *Hug* | 6.64 | 7.02 | 7.61 | 7.38 | 9.27 | 4.3 | 4.4 | **12.1** |
| *StandUp* | **37.35** | 34.23 | 37.01 | 29.16 | 15.01 | 10.1 | 7.6 | 9.0 |
| *SitDown* | 6.51 | 6.95 | 7.53 | 7.40 | **9.06** | 5.3 | 4.2 | 5.6 |
| *Swim* | 16.58 | **29.48** | 17.60 | 29.45 | 26.70 | 11.3 | 5.5 | 7.5 |
| *Dance* | 43.15 | 36.26 | **44.59** | 29.64 | 25.12 | 10.1 | 10.5 | 7.5 |
| **mean AP** | 25.14 | **26.11** | 24.45 | 24.61 | 24.05 | 12.6 | 13.3 | 14.1 |

Table 1: Average precision per class on the Hollywood 3D action dataset. The APs for the bag of words descriptor using the 3D-Ha, 4D-Ha, 3.5D-Ha interest points are reported from [6]. The values in bold are the best AP per class across all methods.

## References

[1] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *J. OPT. SOC. AM. A*, 2(2):284–299, 1985.

[2] C. F. Cadieu and B. A. Olshausen. Learning Intermediate-Level Representations of Form and Motion from Natural Movies. *Neural Computation*, 24(4):827–866, Dec. 2011.

[3] D. Fleet, H. Wagner, and D. Heeger. Neural encoding of binocular disparity: Energy models, position shifts and phase shifts. *Vision Research*, 36(12):1839–1857, June 1996.

[4] D. J. Fleet, A. D. Jepson, and M. R. Jenkin. Phase-based disparity measurement. *CVGIP: Image understanding*, 53(2):198–210, 1991.

[5] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[6] S. Hadfield and R. Bowden. Hollywood 3d: Recognizing actions in 3d natural scenes. In *Proceeedings, conference on Computer Vision and Pattern Recognition*, Portland, Oregon, June23 - 28 2013.

[7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[8] C. Kanan and G. Cottrell. Robust classification of objects, faces, and flowers using natural image statistics. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2472–2479, 2010.

[9] K. R. Konda, R. Memisevic, and V. Michalski. The role of spatio-temporal synchrony in the encoding of motion. *CoRR*, abs/1306.3162, 2013.

[10] Q. Le, W. Zou, S. Yeung, and A. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011.

[11] R. Memisevic. Gradient-based learning of higher-order image features. In *ICCV*, 2011.

[12] R. Memisevic. On multi-view feature learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 161–168, 2012.

[13] I. Ohzawa, G. C. Deangelis, and R. D. Freeman. Stereoscopic depth discrimination in the visual cortex: neurons ideally suited as disparity detectors. *Science*, 249(4972):1037–1041, 1990.

[14] N. Qian. Computing stereo disparity and motion with known binocular cell properties. *Neural Computation*, 6(3):390–404, 1994.

[15] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive Auto-Encoders: Explicit Invariance During Feature Extraction. In *ICML*, 2011.

[16] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[17] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *Proceedings of the 11th European conference on Computer vision: Part VI*, ECCV'10, 2010.