Glimpse Clouds: Human Activity Recognition from Unstructured Feature Points
Fabien Baradel, Christian Wolf, Julien Mille, Graham W. Taylor

To cite this version:

HAL Id: hal-01713109
https://inria.hal.science/hal-01713109
Submitted on 25 Jul 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Glimpse Clouds: Human Activity Recognition from Unstructured Feature Points

Fabien Baradel¹, Christian Wolf¹,², Julien Mille³, Graham W. Taylor⁴,⁵
{fabien.baradel, christian.wolf}@liris.cnrs.fr, julien.mille@insa-cvl.fr, gwtaylor@uoguelph.ca

Abstract

We propose a method for human activity recognition from RGB data that does not rely on any pose information during test time, and does not explicitly calculate pose information internally. Instead, a visual attention module learns to predict glimpse sequences in each frame. These glimpses correspond to interest points in the scene that are relevant to the classified activities. No spatial coherence is forced on the glimpse locations, which gives the attention module liberty to explore different points at each frame and better optimize the process of scrutinizing visual information. Tracking and sequentially integrating this kind of unstructured data is a challenge, which we address by separating the set of glimpses from a set of recurrent tracking/recognition workers. These workers receive glimpses, jointly performing subsequent motion tracking and activity prediction. The glimpses are soft-assigned to the workers, optimizing coherence of the assignments in space, time and feature space using an external memory module. No hard decisions are taken, i.e. each glimpse point is assigned to all existing workers, albeit with different importance. Our methods outperform the state-of-the-art on the largest human activity recognition dataset available to-date, NTU RGB+D, and on the Northwestern-UCLA Multiview Action 3D Dataset.

1. Introduction

We address human activity recognition in settings where activities are complex and diverse, either as performed by an individual, or involving multiple participants. These activities may even include people interacting with objects or the environment. The usage of RGB-D cameras is very popular for this case [44, 34, 56], as it allows for the use of articulated pose (skeletons) to be delivered in real time and relatively cheaply by some middleware. The exclusive usage of pose makes it possible to work on gesture and activity recognition without being a specialist in vision [61, 52], and with significantly reduced dimensionality of the input data. The combined usage of pose and raw depth and/or RGB images can often boost performance over a solution that uses a single modality [41].

We propose a method that only uses raw RGB images at test time. We avoid the use of articulated pose for two reasons: (i) depth data is not always available; for example, in applications involving smaller or otherwise resource-constrained robots; and (ii) the question of whether articulated pose is the optimal intermediate representation for activity recognition is unclear. We explore an alternative strategy, which consists of learning a local representation of video through a visual attention process.

We conjecture that the replacement of the articulated pose modality should keep one important property, which is its collection of local entities, which can be tracked over time and whose motion is relevant to the activity at hand. Instead of fixing the semantic meaning of these entities to the definition of a subset of joints in the human body, we learn it discriminatively. In our strategy, the attention process is completely free to attend to arbitrary locations at each time instant. In particular, we do not impose any constraints on spatio-temporal coherence of glimpse locations, which allows the model to vary its focus within and across frames. Certain similarities can be made to human gaze patterns which saccade to different points in a scene.

Activities are highly correlated with motion [47], and therefore tracking the motion of specific points of visual interest is essential, yielding a distributed representation of the collection of glimpses. Appearance and motion features need to be collected over time from local points and integrated into a sequential decision model. However, tracking a set of glimpse points, whose location is not spatio-temporally smooth and whose semantic meaning can change from frame to frame, is a challenge. Our objective is to match new glimpses with past ones of the same (or a nearby) location in the scene. Due to the unconstrained nature of the attention mechanism, it is not aware of when a point in the scene has been last scrutinized, or if it has been attended to in the past.

¹ Univ. Lyon, INSA-Lyon, CNRS, LIRIS, F-69621, Villeurbanne, France.
² INRIA, CITI Laboratory, Villeurbanne, France.
³ Laboratoire d’Informatique de l’Univ. de Tours, INSA Centre Val de Loire, 41034 Blois, France.
⁴ School of Engineering, Univ. of Guelph, Guelph, Ontario, Canada.
⁵ Vector Institute, Toronto, Ontario, Canada.
We solve this issue by separating the problem into two distinct parts: (i) selecting a distributed and local representation of $G$ glimpse points through a sequential recurrent attention model; and (ii) tracking the set of glimpses by a set of $C$ recurrent workers, which sequentially integrate features and participate in the final recognition of the activity (Figure 1). In general, $G$ can be different from $C$, and the assignment between glimpses and workers is soft. Each worker is potentially assigned to all glimpses, albeit to a varying degree.

We summarize our main contributions as follows:

- We present a method for human activity recognition that does not require articulated pose during testing, and models activities using two attentional processes; one extracting a set of glimpses per frame and one reasoning about entities over time. This model has a number of interesting and general properties:
  
  - This unstructured “cloud” of glimpses produced by the attention process are tracked over time using a set of trackers/recognizers, which are soft-assigned using external memory. Each tracker can potentially track multiple glimpses.
  
  - Articulated pose is used during training time as an additional target, encouraging the attention process to focus on human structures.
  
  - All attentional mechanisms are executed in feature space, which is calculated jointly with a global model processing the full input image.

- We evaluate our method on two datasets, NTU RGB-D and N-UCLA Multiview Action 3D, outperforming the state-of-the-art by a large margin.

2. Related Work

Activities, gestures and multimodal data — Recent gesture and human activity recognition methods dealing with several modalities typically process 2D+T RGB and/or depth data as 3D. Sequences of frames are stacked into volumes and fed into convolutional layers at the first stages [3, 22, 40, 41, 57]. When additional pose data is available [37], the 3D joint positions are typically fed into a separate network. Preprocessing pose is reported to improve performance in some situations, e.g. augmenting coordinates with velocities and acceleration [62]. Fusing modalities is traditionally done as late [40], or early fusion [57]. In contrast, our method does not require pose during testing and only leverages it during training for regularization.

Recurrent architectures for action recognition — Recurrent neural networks (and their variants) are employed in much contemporary work on activity recognition, and a recent trend is to make recurrent models local. Part-aware LSTMs [44] separate the memory cell of an LSTM network [18] into part-based sub-cells and let the network learn long-term representations individually for each part, fusing the parts for output. Similarly, Du et al. [11] use bi-directional LSTM layers that fit an anatomical hierarchy. Skeletons are split into anatomically-relevant parts (legs, arms, torso, etc.) and let subnetworks specialize on them. Lattice LSTMs partition the latent space over a grid that is aligned with the spatial input space [50]. On the other hand, we soft-assign parts over multiple recurrent workers, each worker potentially integrating all points of the scene.

Tracking and distributed recognition — Structural RNNs [21] bear a certain resemblance to our work. They handle the temporal evolution of tracked objects in videos with a set of RNNs, each of which correspond to cliques in a graph that models the spatio-temporal relationships be-
Attention mechanisms and external memory — Attention mechanisms focus selectively on parts of the scene that are the most relevant to the task. Two types of attention have emerged in recent years. Soft attention weights each part of the observation dynamically [4, 25]. The objective function is usually differentiable, allowing gradient-based optimization. Soft attention was proposed for image [9, 58] and video understanding [46, 48, 59] with spatial, temporal, and spatio-temporal variants.

Towards action recognition, Sharma et al. [46] proposed a recurrent mechanism from RGB data, which integrates convolutional features from different parts of a space-time volume. Song et al. [48] proposed separate spatial and temporal attention networks for action recognition from pose. At each frame, the spatial attention model gives more importance to the joints most relevant to the current action, whereas the temporal model selects frames.

Hard attention takes explicit decisions when choosing parts of the input data. In a seminal paper, Mnih et al. [39] proposed visual hard attention for image classification built around an RNN, selecting the next location on based on past information. Similar hard attention was used in multiple object recognition [2], object localization [7, 38, 23], saliency map generation [27], and action detection [60]. While the early hard attention models were not differentiable, implying reinforcement learning, the DRAW algorithm [15] and spatial transformer networks (STN) [20] provide attention crops which are fully differentiable and can thus be learned using gradient-based optimization.

The addition of external memory proved to increase the capacity of neural networks by storing long-term information from past observations; this was mainly popularized by Neural Turing Machines and [14] and Memory Networks [49, 28]. In [1], a Fully Convolutional Network is coupled with an attention-based memory module to perform context selection and refinement for semantic segmentation. In [53], visual memory is used to learn a spatio-temporal representation of moving objects in a scene. Memory is implemented as a convolutional GRU with a 2D spatial hidden state. In [35], the ST-LSTM method of [34] is extended with a global context memory for skeleton-based action recognition. Multiple attention iterations are performed to optimize the global context memory, which is used for the final classification. In [51], an LSTM-based memory network is used for RGB and optical flow-based action recognition.

Our attention process is different from previously published work in that it produces an unstructured Glimpse Cloud in a spatio-temporal cube. The attention process is unconstrained, which we show to be an important design choice. In our work, the external memory module provides a way to remember past soft-assignments of glimpses in the recurrent workers. Furthermore, accessing the external memory is fully-differentiable, which allows for supervised end-to-end training.

3. Glimpse Clouds

We first introduce the following notation: We map an input video sequence $X \in \mathbb{R}^{T \times H \times W \times 3}$ to a corresponding activity label $y$ where $H, W, T$ denote, respectively, the height, the width and the number of time steps. The sequence $X$ is a set of RGB input images $X_t \in \mathbb{R}^{H \times W \times 3}$ with $t=1...T$. We do not use any external information during testing, such as pose data, depth, or motion. However, if pose data is available during training time, our method is capable of integrating it through additional predictions and supervision, which we show increases the performance of the system (Section 4).

Many RGB-only state-of-the-art methods, which do not use pose data, extract features at a frame level by feeding the entire video frame to a pre-trained deep network. This yields global features, which do not capture local information well. Reasoning at a local level has, until now, been achieved using pose features, or attention processes that were limited to attention maps (e.g. [46, 33]). Here, we propose an alternative approach, where an attention process runs over each time instant and over time, creating sequences of sets of glimpse points, from which features are extracted.

Our model processes videos using several key components, as illustrated in Figure 1: i) a recurrent spatial attention model that extracts features from different local glimpses $v_{t,g}$ following an attention path in each video over frames $t$ and multiple glimpses $g$ in each frame; and ii) distributed soft-tracking workers, which process these spatial features sequentially. As the input data is unstructured, the spatial glimpses are soft-assigned to the workers, such that no hard decisions are made at any point. To this end, iii) an external memory module keeps track of the glimpses seen in the past, their features, and past soft-assignments, and produces new soft-assignments optimizing spatio-temporal consistency. Our approach is fully-differentiable, allowing end-to-end training of the full model.

3.1. A joint global/local feature space

We recognize activities jointly based on global and local features. In order to speed up calculations and to avoid extracting redundant calculations, we use a single feature space computed by a global model. In particular, we map an input sequence $X$ to a spatio-temporal feature map $Z \in \mathbb{R}^{H \times W \times 3}$.
using a deep neural network $f(\cdot)$ with 3D convolutions. Pooling is performed on the spatial dimensions, but not on time. This allows for the retention of the original temporal scale of the video, and therefore access to features in each frame. It should be noted, however, that due to the 3D convolutions, the temporal receptive field of a single “temporal” slice of the feature map is greater than a single frame. This is intended, as it allows the attention process to use motion. In an abuse of terminology, we will still use the term frame to specify the slice $Z_t$ of a feature map with a temporal length of 1. More information on the architecture of $f(\cdot)$ is given in Section 5.

3.2. The attention process

Inspired by human behavior when scrutinizing a scene, we extract a fixed number of features from a series of $G$ glimpses within each frame. The process of moving from one glimpse to another is achieved with a recurrent model. Glimpses are indexed by index $g=1\ldots G$, and each glimpse $Z_{t,g}$ corresponds to a sub-region of $Z_t$ using coordinates and scale $I_{t,g} = [x_g, y_g, s_g^x, s_g^y]_T$ output by a differentiable glimpse function, a Spatial Transformer Network (STN) [20]. STN allows the attention process to perform a differentiable crop operation on each feature map. Features are extracted using a transformed ROI average pooling at location $I_{t,g}$, resulting in a 1D feature vector $z_{t,g}$:

$$Z_{t,g} = \text{STN}(Z_t, I_{t,g})$$

$$z_{t,g} = \Gamma(Z_{t,g}) = \frac{1}{H'W'} \sum_m \sum_n Z_{t,g}(m,n)$$

where $W' \times H'$ is the size of the glimpse region. The glimpse locations and scales $I_{t,g}$ for $g=1\ldots G$ are predicted by a recurrent network, which runs over the glimpses. As illustrated in Figure 1, the model predicts a fixed-length sequence of glimpse points for each frame. It runs over the entire video at once, i.e. it is not restarted/reinitialized after each frame. The hidden state therefore carries information across frames and creates a globally coherent scrutiny process for the video. The recurrent model is given as follows (we use GRUs [10] for simplicity’s sake, and we omit gates and biases in the rest of the equations to reduce notational complexity):

$$h_g = \Omega(h_{g-1}, [z_{g-1}, r])$$

$$I_g = W_t \Gamma(h_g, c)$$

where $h$ denotes the hidden state of the RNN running over glimpses $g$, $c$ is a frame context vector for making the process aware of frame transitions (described in Section 3.6), and $r$ carries information about the high-level classification task. In essence, $r$ corresponds to the global hidden state of the recurrent workers performing the actual recognition, as described in Section 3.3 and Equation (7). Note that $h$, $z$, $l$, $r$, and $c$ depend on both the time and glimpse indices. Observing that the recurrence runs over glimpses $g$, the time index $t$ is dropped from Eq. 3–4 for notational simplicity.

3.3. Distributed soft-tracking workers

Extracting motion cues from semantic points in a scene requires associating glimpse points from different frames over time. Due to the freedom of the attention process and fixed number of glimpses, subsequent glimpses of the same point in the scene are generally not in subsequent frames, which excludes the use of conventional tracking mechanisms. Instead, we avoid hard tracking and hard assignments between glimpse points in a temporal manner. We propose a soft associative model for automatically grouping similar spatial features over time.

As given in Equation (2), we denote $z_{t,g}$ as the features extracted from the $g^{th}$ glimpse in feature map $Z_t$ for $g = 1\ldots G$ and $t = 1\ldots T$. We are interested in a joint encoding of spatial and feature dimensions and employ “what” and “where” features $v_{t,g}$, as introduced in [29], defined by:

$$v_{t,g} = z_{t,g} \otimes \Lambda(I_{t,g} | \theta)$$

where $\otimes$ is the Hadamard product and $\Lambda(I_{t,g} | \theta)$ is a network providing an embedding of the spatial patch coordinates into a space of the same dimensionality as the features $z_{t,g}$. The vector $v_{t,g}$ contains joint cues about motion, appearance, and spatial localization.

Evolution of this information over time is modeled with a number $C$ of so-called soft-tracking workers $\Psi_c$, for $c=1\ldots C$, each of which corresponds to a recurrent model capable of tracking entities over time. We never hard assign glimpses to workers. Inputs to each individual worker correspond to weighted contributions from all of the $G$ glimpses. In general, the number of glimpse points $G$ can be different from the number of workers $C$. At each instant,
glimpses are thus soft-assigned to the workers on the fly by changing the weights of the contributions, as described further below.

Workers \( \Psi_c \) are GRUs following the usual update equations based on the past state \( r_{t-1,c} \) and input \( \hat{v}_{t,c} \):

\[
\begin{align*}
  r_{t,c} &= \Psi_c(r_{t-1,c}, \hat{v}_{t,c}, \theta_{\Psi_c}) \\
  r_t &= \sum_c r_{t,c}
\end{align*}
\]  

(6)

(7)

where \( \Psi_c \) is a GRU and \( r_t \) carries global information about the current state (needed as input to the recurrent model of spatial attention). The input \( \hat{v}_{t,c} \) to each worker \( \Psi_c \) is a linear combination of the different glimpses \( \{v_{t,g}\}, g = 1 \ldots G \) weighted by a soft attention distribution \( p_{t,c} = \{p_{t,g,c}\}, g = 1 \ldots G \):

\[
\hat{v}_{t,c} = V_t p_{t,c}
\]

(8)

where \( V_t \) is a matrix whose rows are the different glimpse features \( v_{t,g} \). Workers are independent from each other in the sense that they do not share parameters \( \theta_{\Psi_c} \). This can potentially lead to specialization of the workers on types of tracked and integrated scene entities.

3.4. Soft-assignment using External Memory

The role of the attention distribution \( p_{t,c} \) is to give higher weights to glimpses that have been soft-assigned to worker \( c \) in the past; therefore, workers extract different kinds of features from each other. To accomplish this, we employ an external memory bank denoted \( M = \{m_k\} \), which is common to all workers (see Figure 2). In particular, \( M \) is a fixed-length array of \( K \) entries \( m_k \) each capable of storing a feature vector \( v_{t,g} \). Even if the external memory is common to each worker, they have their own ability to extract information from it. Each worker \( \Psi_c \) has its own weight bank denoted \( W_c = \{w_{c,k}\} \). The scalar \( w_{c,k} \) holds the importance of the entry \( m_k \) for worker \( \Psi_c \). Hence, the overall external memory is defined by the set \( \{M, W_1, \ldots W_C\} \).

Two operations can be performed on the external memory: **reading** and **writing**. Memory reading consists of extracting knowledge that is already stored in memory banks. Meanwhile, memory writing consists of adding a new memory entry to the memory bank. We describe these two fully-differentiable operations below.

**Attention from memory reads** — The attention distribution \( p_{t,c} \) is a distribution over glimpses \( g \), i.e. \( p_{t,c} = \{p_{t,c,g}\} \), \( 0 \leq p_{t,c,g} \leq 1 \) and \( \sum_g p_{t,c,g} = 1 \). We want the glimpses to be distributed appropriately across the workers, and encourage worker specialization. In particular, at each timestep, we want to assign a glimpse of high importance to a worker if this worker has been soft-assigned similar glimpses in the past (also with high importance). To this end, we define a fully trainable distance function \( \phi(\cdot, \cdot) \), which is implemented in a quadratic form:

\[
\phi(x, y) = \sqrt{(x - y) \cdot D(x - y)}
\]

(9)

where \( D \) is a learned weight matrix. Within each batch, we normalize \( \phi(\cdot, \cdot) \) with min-max normalization to scale it between 0 and 1.

A glimpse \( g \) is soft-assigned to a given worker \( c \) with a higher weight \( p_{t,c,g} \) if \( v_{t,g} \) is similar to vectors \( m_k \) from the memory bank \( M \), which had a high importance for the worker in the past \( \Psi_c \):

\[
p_{t,c,g} = \sigma_{\alpha} \left( \sum_k e^{-t \cdot m_k} \times w_{c,k} \left[ 1 - \phi(v_{t,g}, m_k) \right] \right)
\]

(10)

where \( \sigma \) is the softmax function over the \( G \) glimpses and \( e^{-t \cdot m_k} \) is an exponential rate over time to give higher importance to recent feature vectors compared to those in the distant past. \( t \cdot m_k \) is the corresponding timestep of the memory entry \( m_k \). In practice, we add a temperature term \( \alpha \) to the softmax function \( \sigma \). When \( \alpha \to 0 \), the output vector is sparser. The negative factor multiplied with \( \phi \) is justified by the fact that \( \phi \) is initially pre-trained as a Mahalanobis distance by setting \( D \) to the inverse covariance matrix of the glimpse data. The factor therefore transforms the distance into a similarity. After pre-training, \( D \) is trained end-to-end.

The attention distribution \( p_{t,c} \) is computed for each worker \( \Psi_c \). Thus, each glimpse \( g \) potentially contributes to each worker \( \Psi_c \) through its input vector \( \hat{v}_{t,c} \) (c.f. Equation (8)), albeit with different weights.

**Memory writes** — For each frame, the feature representations \( v_{t,g} \) are stored in the memory bank \( M \). However, the attention distribution \( p_{t,c,g} \) is used to weight these entries for each worker \( \Psi_c \). If a glimpse feature \( v_{t,g} \) is stored in a slot \( m_k \), then its importance weight \( w_{c,k} \) for worker \( \Psi_c \) is set to \( p_{t,c,g} \). The only limitation is the size \( K \) of the memory bank. When the memory is full, we delete the oldest memory entry. More flexible storing processes, for example, trained mappings, are left for future work.

3.5. Recognition

Since workers proceed in an independent manner through time, we need an aggregation strategy to perform classification. Each worker \( \Psi_c \) has its own hidden state \( \{r_{t,c}\}_{t=1 \ldots T} \) and is responsible for its own classification through a fully-connected layer. The final classification is done by averaging logits of the workers:

\[
q_c = W_c \cdot r_c
\]

(11)

\[
\hat{y} = \text{softmax} \left( \sum_c q_c \right)
\]

(12)
where \( \hat{y} \) is the probability vector of assigning the input video \( X \) to each class.

### 3.6. Context vector

In order to make the spatial attention process (Section 3.2) aware of frame transitions, we introduce a context vector \( c_t \) which contains high level information about humans present in the current frame \( t \). \( c_t \) is obtained by global average pooling over the spatial domain of the penultimate feature maps of a given timestep. If pose is available at training time, we regress the 2D pose coordinates of humans from the context vector \( c_t \) using the following mapping:

\[
y^p_t = W_p^T c_t
\]

Pose \( y^p_t \) is linked to ground truth pose (only during training) using a supervised term described in Section 4. This leads to hierarchical feature learning in that the penultimate feature maps have to detect human joints present in each frame.

### 4. Training

We train the model end-to-end with the sum of a collection of loss terms, which are explained below:

\[
L = L_D(\hat{y}, y) + L_P(\hat{y}^p, y^p) + L_G(l, y^p)
\]

**Supervision** — \( L_D(\hat{y}, y) \) is a supervised loss term (cross-entropy loss on activity labels \( y \)).

**Pose prediction** — Articulated pose \( y^p \) is available for many datasets. Our goal is to not depend on pose during testing; however, its usage during training can provide additional information to the learning process and reduce the tendency of activity recognition methods to memorize individual elements in the data for recognition. We therefore add an additional term \( L_P(\hat{y}^p, y^p) \), which encourages the model to perform pose regression during training only from intermediate feature maps (described in Section 3.6). Pose regression over time leads to a faster convergence of the overall model.

**Making glimpses similar to humans** — \( L_G(l, y^p) \) is a loss encouraging the glimpses points to be as sparse as possible within a frame, but at the same time, close to humans in the scene. Recall that \( l_{t,g} = [x_{t,g}, y_{t,g}, s_{t,g}, s_{t,g}]^T \), so \( L_G \) is defined by:

\[
L_{G_1}(l_t) = \frac{1}{1 + \sum_{g_1}^{G} \sum_{g_2}^{G} \|l_{t,g_1}, l_{t,g_2}\|}
\]

\[
L_{G_2}(l_t, y^p_t) = \sum_{g}^{G} \min_{j} \|l_{t,g}, y^p_{t,j}\|
\]

\[
L_G(l, y^p) = \sum_{t}^{T} (L_{G_1}(l_t) + L_{G_2}(l_t, y^p_t))
\]

where \( y^p_{t,j} \) denotes the 2D coordinates of joints \( j \) at time \( t \), and Euclidean distance on \( l_{t,g} \) is computed using the central focus point \((x_{t,g}, y_{t,g})\). \( L_{G_1} \) encourages diversity between glimpses within a frame. \( L_{G_2} \) ensures that all the glimpses are not taken too far away from the subjects.

### 5. Pre-trained architecture

We designed the 3D convolutional network \( f(\cdot) \) by computing the global feature maps in Section 3.1, such that the temporal dimension is maintained (i.e. without any temporal subsampling). Using a pre-trained Resnet-50 network [17] as a starting point, we inflate the 2D spatial convolutional kernels into 3D kernels, artificially creating new a temporal dimension, as described by Carreira et al. [8]. This allows us to take advantage of the 2D kernels learned by pre-training on image classification on the Imagenet dataset. The inflated ResNet \( f(\cdot) \) is then trained as a first step by minimizing the loss \( L_D + L_P \). The supervised loss \( L_D \) on the global model is applied on a path attached to global average pooling on the last feature maps, followed by a fully-connected layer that is subsequently removed.

The recurrent spatial attention module \( \Omega(\cdot) \) is a GRU with a hidden state of size 1024; \( \Lambda(\cdot) \) is an MLP with a single hidden layer of size 256 and a ReLU activation; the soft-trackers \( \Psi \) are GRUs with a hidden state of size 512. There is no parameter sharing among them.

### 6. Experimental Results

We evaluated the proposed method on two human activity recognition datasets: NTU RDB+D Dataset [44] and Northwestern-UCLA Multiview Action 3D Dataset [55].

**NTU RDB+D Dataset (NTU)** — NTU was acquired with a Kinect v2 sensor and contains more than 56K videos and 4 million frames with 60 different activities including individual activities, interactions between multiple people, and health-related events. The activities were performed by 40 subjects and recorded from 80 viewpoints. We follow the cross-subject and cross-view split protocol from [44]. Due to the large number of videos, this dataset is highly suitable for deep learning modeling.

**Northwestern-UCLA Multiview Action 3D Dataset (N-UCLA)** — This dataset [55] contains 1494 sequences, covering ten action categories, such as *drop trash* or *sit down*. Each sequence is captured simultaneously by 3 Kinect v1 cameras. RGB, depth and human pose are available for each video, and each action is performed one to six times by ten different subjects. Most actions involve human-object interaction, making this dataset challenging. We followed the cross-view protocol defined by [55], and we trained our method on samples from two camera views, and tested it on samples from the remaining view. This produced three possible cross-view combina-
Figure 3. An illustration of the glimpse distribution for several sequences of the NTU dataset. Here we set 3 glimpses per frame (G=3, Red: first, Blue: second, Yellow: third).

Table 1. Results on the Northwestern-UCLA Multiview Action 3D dataset with Cross-View Setting (accuracy as a percent). V, D, and P mean Visual (RGB), Depth, and Pose, respectively.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Data</th>
<th>V 1,2</th>
<th>V 2,3</th>
<th>V 3,1</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVV [32]</td>
<td>D</td>
<td>58.5</td>
<td>55.2</td>
<td>39.3</td>
<td>51.0</td>
</tr>
<tr>
<td>CVP [64]</td>
<td>D</td>
<td>60.6</td>
<td>55.8</td>
<td>39.5</td>
<td>52.0</td>
</tr>
<tr>
<td>AOG [55]</td>
<td>D</td>
<td>45.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HPM+TM [43]</td>
<td>D</td>
<td>91.9</td>
<td>75.2</td>
<td>71.9</td>
<td>79.7</td>
</tr>
<tr>
<td>Lie Group [54]</td>
<td>P</td>
<td>74.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HBRNN-L [12]</td>
<td>P</td>
<td>78.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Enhanced viz. [36]</td>
<td>P</td>
<td>86.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ensemble TS-LSTM [30]</td>
<td>P</td>
<td>89.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Hankelets [31]</td>
<td>V</td>
<td>45.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>nCTE [16]</td>
<td>V</td>
<td>68.6</td>
<td>68.3</td>
<td>52.1</td>
<td>63.0</td>
</tr>
<tr>
<td>NKTM [42]</td>
<td>V</td>
<td>75.8</td>
<td>73.3</td>
<td>59.1</td>
<td>69.4</td>
</tr>
<tr>
<td>Global model</td>
<td>V</td>
<td>85.6</td>
<td>84.7</td>
<td>79.2</td>
<td>83.2</td>
</tr>
<tr>
<td>Glimpse Clouds</td>
<td>V</td>
<td>90.1</td>
<td>89.5</td>
<td>83.4</td>
<td>87.6</td>
</tr>
</tbody>
</table>

Table 2. Results on the NTU RGB+D dataset with Cross-Subject and Cross-View settings (accuracies in %); (†) indicates method has been re-implemented.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Pose</th>
<th>RGB</th>
<th>CS</th>
<th>CV</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lie Group [54]</td>
<td>✓</td>
<td>-</td>
<td>50.1</td>
<td>52.8</td>
<td>51.5</td>
</tr>
<tr>
<td>Skeleton Quads [13]</td>
<td>✓</td>
<td>-</td>
<td>38.6</td>
<td>41.4</td>
<td>40.0</td>
</tr>
<tr>
<td>Dynamic Skeletons [19]</td>
<td>✓</td>
<td>-</td>
<td>60.2</td>
<td>65.2</td>
<td>62.7</td>
</tr>
<tr>
<td>HBRNN [11]</td>
<td>✓</td>
<td>-</td>
<td>59.1</td>
<td>64.0</td>
<td>61.6</td>
</tr>
<tr>
<td>Deep LSTM [44]</td>
<td>✓</td>
<td>-</td>
<td>60.7</td>
<td>67.3</td>
<td>64.0</td>
</tr>
<tr>
<td>Part-aware LSTM [44]</td>
<td>✓</td>
<td>-</td>
<td>62.9</td>
<td>70.3</td>
<td>66.6</td>
</tr>
<tr>
<td>ST-LSTM + TrustG. [34]</td>
<td>✓</td>
<td>-</td>
<td>69.2</td>
<td>77.7</td>
<td>73.5</td>
</tr>
<tr>
<td>STA-LSTM [48]</td>
<td>✓</td>
<td>-</td>
<td>73.2</td>
<td>81.2</td>
<td>77.2</td>
</tr>
<tr>
<td>Ensemble TS-LSTM [30]</td>
<td>✓</td>
<td>-</td>
<td>74.6</td>
<td>81.3</td>
<td>78.0</td>
</tr>
<tr>
<td>GCA-LSTM [35]</td>
<td>✓</td>
<td>-</td>
<td>74.4</td>
<td>82.8</td>
<td>78.6</td>
</tr>
<tr>
<td>JTM [56]</td>
<td>✓</td>
<td>-</td>
<td>76.3</td>
<td>81.1</td>
<td>78.7</td>
</tr>
<tr>
<td>MTLN [24]</td>
<td>✓</td>
<td>-</td>
<td>79.6</td>
<td>84.8</td>
<td>82.2</td>
</tr>
<tr>
<td>VA-LSTM [63]</td>
<td>✓</td>
<td>-</td>
<td>79.4</td>
<td>87.6</td>
<td>83.5</td>
</tr>
<tr>
<td>View-invariant [36]</td>
<td>✓</td>
<td>-</td>
<td>80.0</td>
<td>87.2</td>
<td>83.6</td>
</tr>
<tr>
<td>DSSCA - SSLM [45]</td>
<td>✓</td>
<td>✓</td>
<td>74.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>STA-Hands [5]</td>
<td>X</td>
<td>X</td>
<td>82.5</td>
<td>88.6</td>
<td>85.6</td>
</tr>
<tr>
<td>Hands Attention [6]</td>
<td>✓</td>
<td>✓</td>
<td>84.8</td>
<td>90.6</td>
<td>87.7</td>
</tr>
<tr>
<td>C3D†</td>
<td>-</td>
<td>✓</td>
<td>63.5</td>
<td>70.3</td>
<td>66.9</td>
</tr>
<tr>
<td>Resnet50+LSTM†</td>
<td>-</td>
<td>✓</td>
<td>71.3</td>
<td>80.2</td>
<td>75.8</td>
</tr>
<tr>
<td>Glimpse Clouds</td>
<td>-</td>
<td>✓</td>
<td>86.6</td>
<td>93.2</td>
<td>89.9</td>
</tr>
</tbody>
</table>

6.1. Implementation details

Similar to [44], we cut videos into sub-sequences of 8 frames and sample sub-sequences. During training, a single sub-sequence is sampled. During testing, 5 sub-sequences and logits are averaged. RGB videos are rescaled to 256 × 256 and random cropping of size 224 × 224 is done during training and testing.

Training is performed using the Adam Optimizer [26] with an initial learning rate of 0.0001. We use minibatches of size 40 on 4 GPUs. Following [44], we sample 5% of the initial training set as a validation set, which is used for hyper-parameter optimization and for early stopping. All hyperparameters have been optimized on the validation sets of the respective datasets. We used the model trained on NTU as a pre-trained model and fine-tuned it on N-UCLA.

6.2. Results

Comparison with the state of the art — Our method outperforms state-of-the-art methods on NTU and N-UCLA by a large margin, and this also includes several methods which use multiple modalities, in addition to RGB, depth and pose. Tables 1 and 2 provide detailed results comparing to the state-of-the-art on the NTU dataset. Sample visual results can be seen in Figure 3.

Ablation study — Table 3 shows several experiments to study the effect of our design choices. Classification from the Global Model alone (Inflated-Resnet-50) is clearly inferior to the distributed recognition strategy using the set of workers (+1.9 points on NTU and +4.4 points on N-
different outputs of the location network. This is not a full
model, which predicts glimpse locations in parallel through
the glimpses is clearly important; and (ii) with a saliency
drop of more than 6 points, indicating that the location of
plete random sampling of local features, which leads to a
sampling of local features in the space-time volume. We
formance, or whether the gain is solely explained from the
whether a sequential attention process contributes to per-
characteristic, and therefore higher efficiency of a local representation.
The additional loss predicting pose during training helps,
even though pose is not used during testing. An important
question is whether the Glimpse Cloud could be inte-
grated with an easier mechanism than a soft-assignment.
We tested a baseline which sums glimpse features for each
time step and which integrates them temporally (row #3).
This gave only a very small improvement over the global
model. Distributed recognition from Glimpse Clouds with
soft-assignment clearly outperforms the simpler baselines.
Adding the global model does not gain any importance.

**Importance of losses** — Table 3 also shows the relative
importance of our three loss functions. Cross-entropy only
$L_D$ gives 89.1%. Adding pose prediction $L_P$ we gain 0.6
points and adding pose attraction $L_G$ we gain 0.4 points,
which are complementary.

**Unstructured vs. coherent attention** — We also eval-
uated the choice of unstructured attention, i.e. the decision
to give the attention process complete freedom to attend to
a new (and possibly unrelated) set of scene points in each
frame. We compared this with an alternative choice, where
glimpses are axis-aligned space-time tubes over the whole
temporal length of the video. In this baseline, the attention
process is not aligned with time. At each iteration, a new
tube is attended in the full space-time volume, and no track-
or soft-assignment to worker modules is necessary. As
indicated in Table 4, this choice is sub-optimal. We conjec-
ture that tubes cannot cope with moving objects and object
parts in the video.

**Attention vs. saliency vs. random** — We evaluated
whether a sequential attention process contributes to per-
formance, or whether the gain is solely explained from the
sampling of local features in the space-time volume. We
compared our choice with two simple baselines: (i) com-
plete random sampling of local features, which leads to a
drop of more than 6 points, indicating that the location of
the glimpses is clearly important; and (ii) with a saliency
model, which predicts glimpse locations in parallel through
different outputs of the location network. This is not a full
attention process in that a glimpse prediction does not de-
pend on what the model has seen in the past. This choice is
also sub-optimal.

**Learned weight matrix** — Random initialization and
fine-tuning of $D$ matrix in Equation 9 loses 0.4 points and
leads to slower convergence by a factor of 1.5. Fixing $D$ (to
inverse covariance) w/o any training loses 0.8 points.

**The Joint encoding** — “What and where” features are
important for correctly weighting their respective contribu-
tion. Plainly adding concatenating coordinates and features
loses 1.1 points.

**Hyper-parameters $C$, $G$, $T$** — Number of glimpses
and workers: $C$ and $G$ were selected by cross-validation on
the validation set by varying them from 1 to 4, giving an
optimum of $G=C=3$ over all 16 combinations. More leads
the model to overfit. The size of the memory bank $K$ is set
to $T$ where $T=8$ is the length of the sequence.

**Runtime** — The model has been trained using data-
parallelism over 4 Titan Xp GPUs. Pre-training the global
model on the NTU dataset takes 16h. Training the Glimpse
Cloud model end-to-end takes a further 12h. A single for-
ward pass over the full model takes 97ms on 1 GPU. The
method has been implemented in PyTorch.

7. Conclusion

We proposed a method for human activity recognition that
does not rely on depth images or articulated pose, though it
is able to leverage pose information available during train-
ing. The method achieves state-of-the-art performance on
the NTU and N-UCLA datasets even when compared to
methods that use pose, depth, or both at test time. An atten-
tion process over space and time produces an unstructured
Glimpse Cloud, which is soft-assigned to a set of track-
ing/recognition workers. In our experiments, we showed
that this distributed recognition outperforms a global con-
volutional model, as well as local models with simple base-
lines for the localization of glimpses.

**Acknowledgements** — This work was funded by grant
Deepvision (ANR-15-CE23-0029, STPGP-479356-15), a
joint French/Canadian call by ANR & NSERC.
References


[58] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In ICML, pages 2048–2057, 2015. 3


