Discovering the perceptual signatures of Joint Attention

Guido Pusiol, Michael Frank, Fei-Fei Li, Laura Soriano

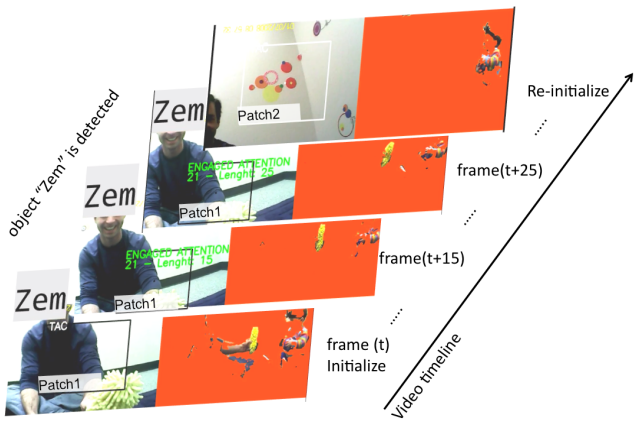
Stanford .. Blah Blah

Abstract

Keywords: Joint attention. Computer vision. Children development.

Introduction

What is it like to be a baby? Lab experiments allow researchers to ask focused questions about childrens social development, but experiments often leave us ignorant about childrens social input in their day-to-day life. Recent technical improvements such as light-weight cameras offer the opportunity to go beyond this previous work and measure what children actually see, but analyzing idiosyncratic records of a childs experience is at best slow and labor-intensive. Our project aims to develop robust tools for analyzing video data from the childs first person perspective.



(a) todo

Figure 1: todo...

An important and not yet fully formalized child-caregiver interaction comes in the shape of joint attention. Joint attention can be seen as the event of two individuals sharing the attention into a third object with mutual awareness of that fact. The detection of a joint attention episode is a strong indicator of learning. Nevertheless, the mutual awareness is perceptually hard for humans to observe. We are addressing the problem in a data driven fashion. Our bottom up approach aims at combining different indicators of mutual awareness to detect joint attention. The goal of this paper is the mining of the importance of different features combinations for Joint attention recognition.

The method uses synchronized head mounted and fixed camera videos and it is composed of different stages.

The first stage, aims at detecting different type of features (i.e. **faces, objects**) extracted frame by frame in the fixed camera information. The objects are the mutually attended ones. In general these objects are being handled and therefore they are moving. For each video frame our algorithm computes a motion mask and tracks the moving object by its color histogram. Over the extracted motion mask the object of interest is tracked using a Mean-Shift based color tracker. Another feature are attention chunks, which are the segments of the childs head mounted video attending visually to the same concept (object, person, etc.). We detect an attention chunk by tracking a similar pixel texture over consecutive video frames.

The second stage, combines the previous features as input of an automatic learning algorithm that can describe the salient combinations for the detection of joint attention. The algorithm can be seen as a deep learning technique that builds a two layer network of feature clusters.

Features Computation

1. Face Detection. Traditional off-the-shelf face detection algorithms fall short to detect the parent faces in the baby-cam dataset. Face detectors work accurately when the test dataset present low variability on the testing dataset and the distance is above 1 meter (e.g. Facebook like pictures). In the baby perspective other type of face configurations appear. Faces appear partially, blurred and with big size and texture variability (fig. **??**) making their detection challenging. We address the problem by with a semi automated adaptive algorithm (?, ?). The algorithm requires manual user input (selecting a face example per video) for its initialization, but then needed no additional training data. The algorithm uses new pixel patches in the trajectory of an optical-flow based tracker to train and update a face detector. The optical flow tracker and the face detector work in parallel. If the face detector finds a location in a new frame exhibiting a high similarity to its stored template, the tracker is re-initialised on that location. Otherwise, the tracker uses the optical flow to decide the location of a face in the new frame. The primary advantage of the algorithm is the use of motion for face detection: Following the movement of the pixels that define a face it is possible for the algorithm to adapt to new morphologies (i.e. different face poses). **Precision.** We evaluate the face detector using 37 videos of children's head mounted cameras. Our evaluation compares the automatically detected faces with human ground truth annotations. The metrics: precision = .92; recall = .99; true Negatives = 93; accuracy = 96 outperform the

(a) Blurred (b) Partial face (c) Low texture

Figure 2: Results (red rectangles) of our face detector in difficult frames.

state of the art (?, ?) due to the active-learning stage of the algorithm.

2. Object detection and tracking. An important cue of activity is related to motion. The detection of objects being handled even when they are not fully in the visual scope of the children could indicate that the joint attention is established. To automatically track and detect moving objects we first need to compute the foreground of the video.

2.1. Background Modeling is the most important part of any foreground subtraction algorithms. The goal is to construct and maintain a statistical representation of the scene to be modeled. Here, we chose to utilize both texture information and color information when modeling the background. The approach (?, ?) exploits the Local Binary Pattern (LBP) feature as a measure of texture because of its good properties (?, ?), along with an illumination invariant photometric distance measure in the RGB space.

The background model $B^t(x)$ of the pixel x at the time t is represented by a list of modes $\{m_k^t(x)\}_{k=1...K}$. The modes register historic information of 7 features of the pixels color and surrounding texture. $m_k^t = \{I, \hat{I}, \check{I}, LBP_k, w, \hat{w}, P\}$. Where I represents the average RGB image vector. $\hat{I}$ and $\check{I}$ are the maximal and minimal RGB image vectors. LBP is the vector of local binary patterns computed at this mode.

The background model can learn up to K^{max} different modes. For each new I^t and LBP^t a new mode is computed m_k^{t-1} and the algorithm seeks to which learned mode m_k^{t-1} the new mode maps to. The mapping is achieved by thresholding a distance (i.e $\tilde{k} = \arg\min_k D(m_k^{t-1}, m_k^t)$). 1) If the new mode cannot be mapped to any of the learned ones and there is still space in the buffer ($K<K^{max}$), then a new mode is initialized. 2) If there is a matched mode $m_{\tilde{k}}^{t-1}$, its representation is updated as follows:

$$\begin{cases} \check{I}_{\tilde{k}}^t = min(I^t, (1+\beta)\check{I}_{\tilde{k}}^{t-1}), \\ \hat{I}_{\tilde{k}}^t = max(I^t, (1-\beta)\hat{I}_{\tilde{k}}^{t-1}), \\ I_{\tilde{k}}^t = (1+\alpha)I_{\tilde{k}}^{t-1} + \alpha I^t, \\ LBP_{\tilde{k}}^t = (1+\alpha)LBP_{\tilde{k}}^{t-1} + \alpha LBP^t, \\ (*)w_{\tilde{k}}^t = (1-\alpha_w^i)w_{\tilde{k}}^{t-1} + \alpha_w^i, \\ \qquad \alpha_w^i = \alpha_w(1+\tau\hat{w}_k^{t-1}) \\ \hat{w}_{\tilde{k}}^t = max(\hat{w}_{\tilde{k}}^{t-1}, w_{\tilde{k}}^t) \\ L_{\tilde{k}}^t = 1 + max\{L_k^{t-1}\}_{k=1,...,K,k\neq\tilde{k}}, \\ \qquad if \;\; L_{\tilde{k}}^t = 0 \; and \; T_{bw} < \hat{w}_{\tilde{k}}^t \end{cases} \quad (1)$$

where $\beta \in [0,1)$ is the learning rate of the min and max color vectors and $\alpha \in [0,1)$ is the learning rate of the color and texture information. The non matching modes of the previous model are assigned to the new model (i.e. $m^t k := m^{t-1} k$) but their weights are decreased according to (*).

After the update step, all nodes are sorted decreasingly according to their weight. And the background modes are the first B^t modes that satisfy:

$$\sum_{k=1}^{B^t} w_k^t / \sum_{k=1}^{K^t} w_k^t < TB \quad (2)$$

where $TB \in [0,1]$ is the background threshold. Note that the use of both color and texture, the chances that moving foreground objects generate a consistent mode overtime (and beneciate from this effect) are quite small.

2.2 Object tracking. We are interested in detecting a tracking a set of objects (i.e. toys) components of triad when the parents and children are engaged in a JA episode. In particular the toys are being handled. The detection and tracking objects is performed over the foreground image. From a computer vision perspective, the objects are highly deformable making them hard to detect and track. The deformations are due to the changes of positions and hand-object occlusions that the object can take while is being manipulated. We have tried different appearance-based (?, ?) object detectors, and all of them failed. We have finally adopted to detect and track them by its color and relative size. We modified the cam-shift algorithm (?, ?), which is a specialization of the well known mean-shift algorithm (?, ?). The mean shift algorithm is a non-parametric technique that climbs the gradient of a probability distribution to find the nearest dominant mode (peak). In our case the distribution is based in color values. The algorithm initializes selecting a region containing the object of interest, and building a color histogram over the region. In a new frame, the algorithm will match the region size and the peaks of the color distribution using mean-shift and the euclidean distance. The figure **??** depicts examples of the detection and tracking of two different objects.

2.3. Attention Chunks: The attention chunks capture the segments of video of a child attending visually to the same

concept (object, person, etc.). The ideal case is to have eye trackers to measure the gaze of the baby, but such a configuration is very hard to achieve in uncontrolled environments. Thus, we use the head-cam information as a gaze estimator. Our approach is backed by psychophysical experiments that indicate eye gaze and head pose are coupled in various tasks (?, ?, ?). The attention chunk is computed automatically on the head camera video. The algorithm is initializes by modeling a pixel-texture patch (P_i). For each new frame the algorithm will seek for a similar patch to the one observed in the previous frame. If the patch is matched, a new point is added to the patch trajectory. If the matching is not achieved, a new patch (P_{i+1}) is learned and the tracking algorithm is re-initialized. The base algorithm used for tracking is a version of a tracker by detection algorithm (?, ?). An Attention Chunk is the video segment defined by the $start_P$ and end_P frames of the tacked patch trajectory. The figure **??** describes the basics of an attention chunk computation.

We perform analytics over the obtained chunk lengths to understand behavioral structures hidden in the data. We have categorized the extracted chunks by their duration in groups of children of different ages. We used 37 children head can videos (20 min. each) grouped by age: 6, 12 and 16 month old. The average duration of the chunks is depicted in the figure **??**. Younger children show to have longer episodes of attending to a spot. While the reasons will require of further discussion, our thoughts are that 16 month old children can walk and are more autonomous, and can avoid the parents imposition of attending to certain objects. The figure **??** depicts the temporal duration of the chunks distributed in buckets of 10ms in a log scale. The graph shows that the decay is proportionally symmetric among the different groups, most of the attention chunks are of short duration and there are not "distractors" (i.e. a TV) that interfere with our guided attention towards new objects.

3. Aggregation: Mapping all features together

In the previous sections we detected and computed features (faces, objects, etc.) in different cameras. The goal of this stage is to merge and prune the features into a single feature matrix describing the detected features at each video frame.

3.1. Synchronization We calculate the bijective function mapping each frame of the fixed camera to the head mounted camera (i.e. figure). When the inter camera frames can be mapped, the detected features and video attributes of both cameras can also be mapped together.

3.2. Feature propagation The features that are detected in a single video frame occupied by an attention chunk are propagated to all of the frames of the chunk. This step improves the quality of the detection assuming that the attention chunks are strong indicators of the child engaged in a full attention episode.

3.3. Binarization The features are binarized for normalization purposes. The features representing a detection (i.e. face and objects) event, are binary by definition. The features representing motion and length are binarized using the median of all observed values as their threshold.

Mining the Features

Our training set is an array of feature vectors $F = \{x^{(1)} \dots x^{(m)}\}$. Each $x^{(i)}$ represents a binary code of the detected features. We feed F to a two layer clustering network composed in the following way:

In the **first layer**, we learn a dictionary $D_a \in \mathbb{R}^{n \times k}$ of k vectors so that a data vector $x^{(i)} \in \mathbb{R}^n$ where $i = 1, \dots, m$ can be associated to a code vector $s^{(i)}$:

$$\begin{aligned} \underset{D_a, s}{\text{minimize}} \quad & \sum_i ||D_a s^{(i)} - x^{(i)}||_2^2 \\ \text{subject to} \quad & ||s^{(i)}||_0 \leq 1, \forall i \\ \text{and} \quad & ||D_a^{(j)}||_2 = 1, \forall j \end{aligned}$$

where $D_a^{(j)}$ is the j'th column of D_a. The **second layer** takes as input the columns of D_a for $a = 1, \dots, n$ training agents. We learn a new dictionary $D \in \mathbb{R}^{2 \times k}$ of k vectors so that a data vector $D_a^{(i)} \in \mathbb{R}^2$ where $i = 1, \dots, k$, and $a = 1, \dots, n$ can be mapped to a code vector $s^{(i)}$:

$$\begin{aligned} \underset{D_a, s}{\text{minimize}} \quad & \sum_i ||D s^{(i)} - D_a^{(i)}||_2^2 \qquad a = 1, \dots, n \\ \text{subject to} \quad & ||s^{(i)}||_0 \leq 1, \forall i \\ \text{and} \quad & ||D^{(j)}||_2 = 1, \forall j \end{aligned}$$

The columns of D are the k hyper-features composing describing the existence or not of an episode of joint attention.

Evaluation

First order analytics can be extracted of the alignment of the automatic feature detection and the annotations. We map into the video timeline the video segments annotated as JA episode and the face and moving object detection. From the alignment, we compute the percentage of time of the automatic detection occurring inside and outside the manually annotated JA episode. We have performed this for a total 77,825 video frames of 4 different children. Table

Dataset

Evaluation Metrics

Results

Conclusions

References

Bradski, G. R. (1998). Computer Vision Face Tracking For Use in a Perceptual User Interface.

	First order feature analysis				
	%Faces IN	%Faces OUT	%Toy IN	% Toy OUT	Total frames
Child 0807	19%	4%	85%	$< 0\%$	19284
Child 0815	21%	5%	51%	$< 0\%$	22421
Child 1604	29%	16%	44%	$< 0\%$	19337
Child 0801	8%	4%	11%	11%	16783

Table 1: The table shows the percentage (normalized by time) of detected faces and handled object inside and outside a JA episode. The distribution of faces and handled object is dense inside the JA episode showing that those features are indicators of a JA episode.

Comaniciu, D., & Meer, P. (2002, May). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(5), 603–619.

Heikkila, M., & Pietikainen, M. (2006, April). A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(4), 657–662.

Kalal, Z., Matas, J., & Mikolajczyk, K. (2010, June). P-N learning: Bootstrapping binary classifiers by structural constraints. *IEEE Conference on Computer Vision and Pattern Recognition. Proceedings*, 49–56.

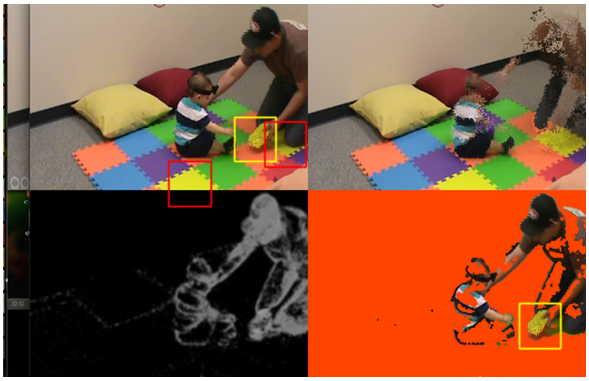
Land, M. F., & Hayhoe, M. (2001). In what ways do eye movements contribute to everyday activities? *Vision research*.

Pelz, J. B., & Canosa, R. (2001). Oculomotor behavior and perceptual strategies in complex tasks. *Vision research*.

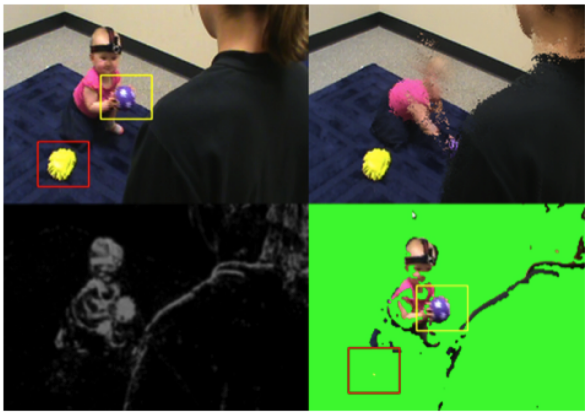
Robust Object Tracking Based on Tracking-Learning-Detection. (2012, May). , 1–60.

Yao, J., & Odobez, J. M. (2007). Multi-layer background subtraction based on color and texture. *... and Pattern Recognition*.

Zhu, X., & Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *Cvpr* (pp. 2879–2886). IEEE.



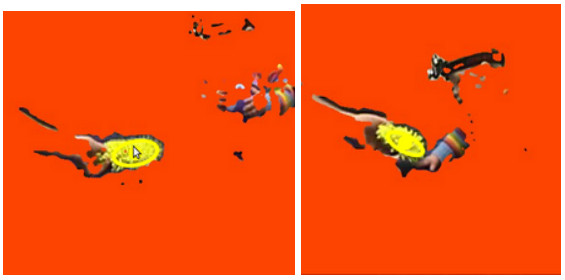
(a) Object of interest (i.e. yellow square) could be confused with other similar textures and color objects in the scene (i.e red squares). The background mask (bottom-right) is filtering the problems out.



(b) Objects of interest are being handled and therefore are moving (i.e. yellow square). Forgotten objects (i.e red squares) are filtered out by the foreground mask.

Figure 3: Foreground computation. In each image, top-left: original color frame, top-right: background model color vector (I), bottom-left: foreground weights, bottom-right: foreground extraction (i.e. orange or green pixels correspond to the background mask). (a) depicts the problem of similar texture and color of the scene and the object of interest. (b) represents the problem of uninteresting objects visible in the scene. Both problems are addressed at once with background/foreground segmentation filtering.

(a) Green object is detected



(b) Zem is detected

Figure 4: (a) the green object is detected, its size and momentum is depicted by the yellow ellipse. (b)

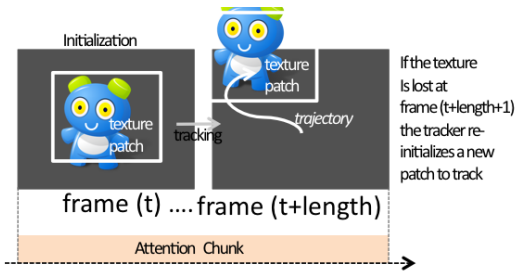
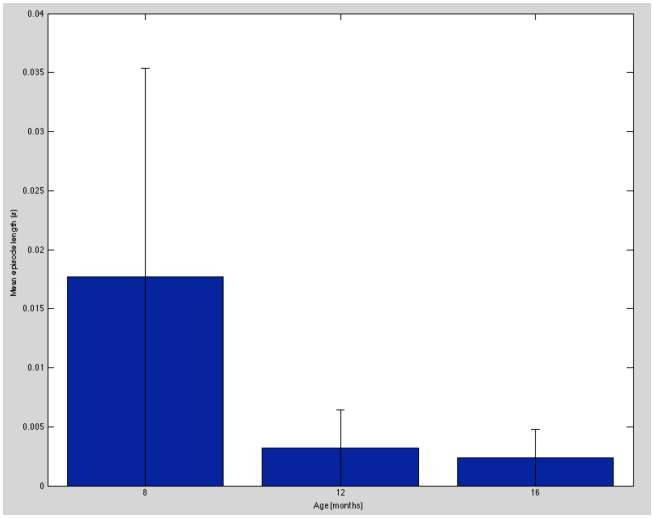
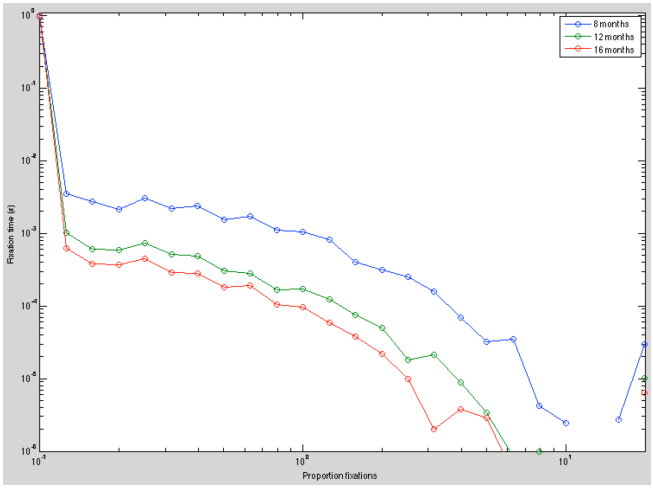


Figure 5: Attention chunk computation.



(a) Duration of attentional chunks ordered by children age 8,12 and 16 month old.



(b) Duration of the attentional chunks distributed in buckets of 10ms and displayed in log scale (i.e. longer chunks appear to the right).

Figure 6: Attention Chunks: analyzing the duration in children of 8,12 and 16 month old.