**Introduction**

So the Topic of the presentation is about spatial Cognition process where we are going to explore about How the computer vision systems will understand these real world assembly process.

So lets first see what is spatial cognition and what is its importance.

Spatial cognition refers to the mental processes through which individuals perceive, comprehend, and navigate the physical space around them. It encompasses a range of cognitive abilities, such as perceiving and interpreting spatial information, remembering spatial layouts, and making decisions based on spatial relationships. This complex process plays a fundamental role in our everyday lives, as it allows us to interact with our environment effectively. By understanding spatial cognition, we can develop improved computational tools and processes that cater to our spatial needs, benefiting fields like architecture, psychology, and neuroscience.

* Architects and interior designers use spatial cognition principles to optimize building layouts, create efficient spaces, and design interiors that enhance user experience and usability.
* Psychology and neural sciences are field of studies which are backed up by spatial cognition. Some of the technologies that are leveraged by these communities are
  + AR/VR : Heavily relies on spatial cognition and even many games like pokemon GO and daily to use technologies like navigation apps such as google maps use this.

By unraveling the complexities of spatial cognition, we gain valuable insights into how humans interact with and understand the world around them.

In This we explore the area which the last research paper stopped. In that paper we saw how the experiment is carried their procedure and set up.

This paper presents a method for conducting fine-grained structural parsing of time series data derived from spatial assemblies. The approach involves several key steps, starting with an exploration of related work in the field. Building on this foundation, the paper proceeds to define a representation for spatial assemblies, which includes specifying an action that can modify them. Next, the authors derive a probabilistic model that captures the underlying assembly processes, along with developing efficient inference algorithms to make sense of the data. By following this systematic flow, the paper aims to contribute to the understanding and analysis of spatial assemblies using time series data, offering valuable insights into the dynamics of these complex systems

Related Works

This paper presents a method for conducting fine-grained structural parsing of time series data derived from spatial assemblies. The approach involves several key steps, starting with an exploration of related work in the field. Building on this foundation, the paper proceeds to define a representation for spatial assemblies, which includes specifying an action that can modify them. Next, the authors derive a probabilistic model that captures the underlying assembly processes, along with developing efficient inference algorithms to make sense of the data. By following this systematic flow, the paper aims to contribute to the understanding and analysis of spatial assemblies using time series data, offering valuable insights into the dynamics of these complex systems.

Action recognition from procedural data involves fine-grained tasks where an agent aims to modify the state of an entity to achieve a desired outcome. This type of recognition often relies on procedural data, such as recipes, which describe step-by-step processes. A recent research paper in the field of natural language processing explores the use of procedural text data to predict state changes resulting from actions. The paper emphasizes the parsing of state changes over time, which inherently addresses action recognition. Traditionally, procedural datasets have been utilized for segmenting and classifying actions, but this paper takes a novel approach by leveraging recipe data to predict state changes. By parsing these changes, the model implicitly captures the recognition of actions. This research contributes to advancing our understanding of action recognition from procedural data, opening up new possibilities for applications in various domains.

Representation

Probabilistic Graphical Model

Motivation

So we are going to use probabilistic graphical model because it provides a structured and principle framework for representing and reasoning about complex assembly processes. Furthermore it is to capture dependencies and uncertainties between variables in assembly processes. This is important because real-world assembly processes can be complex and involve multiple interacting components. By using a graphical model, they were able to represent the relationships between different variables, such as the observed images, the poses of the blocks, and the states of the assembly. And the main reason behind using it is to enable efficient inference algorithms for estimating the state sequence and joint probability of observed and inferred variables [2]. This is crucial for understanding assembly processes from video data, as it allows for the estimation of the underlying states and their transitions over time.

The model presented in this research is defined using a time series structure inspired by a partially-observable Markov decision process (POMDP).

Why POMDP?

The POMDP framework is well-suited for modeling tasks where the agent's knowledge about the underlying state is incomplete or uncertain. In the context of assembly processes, the agent may not have complete information about the current state of the assembly due to occlusions or limited sensor capabilities. By using a POMDP, the authors can capture the partial observability of the assembly process and make informed decisions based on the available information.

In this context, the model aims to estimate the states from a sequence of video keyframes. While traditional POMDPs are concerned with optimizing actions based on a reward function, the focus here is on recognizing the sequence of states resulting from actions taken by an agent. By employing the POMDP framework within the time series structure, the model enables the analysis and estimation of states in the given video data, contributing to advancements in understanding and interpreting temporal information.

Observation model

The graph-based representation employed in this research is effective in differentiating between assemblies with varying structures. However, to generate an image, additional specifications are needed. Specifically, the global pose (pi) of each sub-part or connected component in the state graph must be determined. By considering appearance, shape, and known camera parameters, a template (T) can be created by rendering each sub-part using the poses in (pi). Disconnected sub-parts can be rendered independently, disregarding collisions and occlusions. To account for the errors introduced during rendering, each pixel in the observed image is modeled as an independent Gaussian random variable. The joint probability of the image (I) generated by a given state (s) and pose (p) is expressed using a probabilistic model, considering the pixel values and variances. The orientation and translation of each sub-part's pose distribution are assumed to be uniformly distributed on the unit sphere and within a bounded volume visible by the camera, respectively. By incorporating these observation models, the research aims to enhance the understanding and representation of the assembly process through image generation and analysis.

Process Model

The process model in this paper refers to the probabilistic model that captures the dynamics of the assembly process. It represents the relationships between the states of the assembly, the actions taken by the agent, and the observed images

In an assembly process, the state (st) undergoes construction or deconstruction through the actions of an agent. When only partial information about the actions taken is available, the state transition probabilities can be constructed using the available information. By combining the state transition probabilities (P(st|st-1, at-1)) and the action probabilities (P(at-1|st-1)), the transition probabilities are computed.

in cases where such information is unavailable, the assembly process can be treated as a Markov chain, and the state transition distribution can be directly estimated.

By combining this with the observation model the joint probability of all observed and inferred variables is obtained.

This joint probability is represented by a graphical model, as depicted in Figure

So what did we achieve by doing this?

By incorporating both the process model and the observation model, a comprehensive understanding of the assembly process can be achieved, enabling accurate inference of the observed and inferred variables.

Inference

The inference part of the process model involves estimating the state sequence of the assembly given the observed images. This is done using a Viterbi-style decoding algorithm on the graphical model

The goal is to find the most probable state sequence, which maximizes the joint probability of the observed images, the state sequence, and the poses of the blocks.This can be formulated as the shown optimization problem

where p\* (1:T) and s\* (1:T) represent the optimal state sequence and pose sequence, respectively, and P(I(1:T), p(1:T), s(1:T)) is the joint probability of all observed and inferred variables.

To solve this problem, the paper suggests utilizing max-sum message passing,

Max-sum message passing is a belief propagation algorithm that is used to specifically factor graphs. It is based on the max-sum algorithm, which aims to find the most probable assignment of values to variables in the graph.

In max-sum message passing, messages are passed between the nodes of the graph, representing the beliefs or probabilities associated with each variable. The messages are computed by maximizing over the neighboring variables, taking into account the incoming messages from other nodes.

[The algorithm iteratively updates the messages until convergence, ensuring that each node has the most up-to-date belief based on the information from its neighbors. The final beliefs can then be used to make inferences or decisions based on the graph structure and the observed evidence.]

the hypothesize-and-test approach is used to estimate the state sequence of the assembly given the observed images. At each time step, multiple hypotheses about the assembly state are generated based on the available information and the model assumptions.

The hypothesize-and-test approach is a problem-solving strategy that involves generating multiple hypotheses or possible solutions to a problem and then testing each hypothesis to determine its validity or likelihood. This approach is commonly used in various fields, including scientific research, engineering, and artificial intelligence.

These hypotheses are then evaluated or tested locally by rendering a template and registering it to the observed image. The evaluation involves comparing the rendered template with the observed image to determine the likelihood or fit of each hypothesis.

The evaluation process takes into account various factors, such as the similarity between the rendered template and the observed image, the consistency with the observed poses of the blocks, and the available information about the actions taken by the agent.

After evaluating the hypotheses locally, the inference algorithm proceeds to decode globally by choosing the most probable sequence of hypotheses according to the model. This is done using the Viterbi algorithm with beam search, which efficiently selects and decodes hypotheses based on their probabilities.

Why we are doing it? By iteratively generating and testing hypotheses, the hypothesize-and-test approach allows the inference algorithm to explore different possibilities and refine its estimate of the state sequence based on the observed evidence.

Hypothesis generation

Template Registration

Template registration is used to evaluate the fit or likelihood of each hypothesis by comparing a rendered template to the observed image.

Template registration involves aligning a rendered template of the assembly to the observed image by estimating the best pose (rotation and translation) for the template . The goal is to find the pose that maximizes the similarity between the rendered template and the observed image.

The optimization problem for template registration is formulated as minimizing the sum-of-squared-errors (SSE) distance metric between the rendered template and the observed image. The SSE distance measures the discrepancy between the pixel values of the template and the corresponding pixels in the observed image.

To solve the optimization problem, the Trust Region Reflective algorithm implemented in SciPy is used . This algorithm iteratively adjusts the pose parameters of the template to minimize the SSE distance.

The template registration step is performed for each sub-part or block in the assembly . The algorithm initializes the pose at the centroid of each image segment classified as a block and samples uniformly-spaced values on the unit circle for rotation. The SSE objective is computed, taking into account any missing data, such as pixels classified as hands, and the best assignment of components to image segments is determined using the Hungarian algorithm for assemblies with multiple connected components.

By performing template registration for each sub-part, the algorithm evaluates the fit of each hypothesis by comparing the rendered template to the observed image. This evaluation provides a measure of the likelihood or probability of each hypothesis given the observed evidence.

Parsing Block Construction Process

the authors apply the algorithm discussed to the task of parsing videos of DUPLO block building activity. The goal is to automatically analyze and understand the construction processes involved in building block structures.

The authors collected a dataset consisting of video and inertial measurement data from 30 experimental trials [1]. They performed five examples of each of the six target models shown in Figure 5, ensuring that there was an unobstructed view of each partial assembly as it was being created.

The parsing of block construction processes involves several steps,

First, image pre-processing is performed to remove the background and segment the relevant objects in the frame [1]. Background subtraction is used to fit a plane to the depth image and mask out pixels within a certain distance from the plane. Additionally, the left-most portions of each image, which often contain distracting objects, are masked out. Semantic segmentation is then applied to classify pixels as belonging to blocks or hands using a color-space segmentation and majority voting approach.

The results of the parsing process are evaluated on the collected dataset [1]. The performance of the system is assessed based on how well it can parse the block construction processes and accurately estimate the state sequence.

Evaluation

the authors evaluate the performance of their system on two datasets: the controlled dataset and the child's play dataset. The evaluation is conducted to assess the accuracy, precision, and recall of the system.

In the controlled dataset evaluation, the system is trained using the full child dataset and tested on the full controlled dataset [1]. The state accuracy, precision, and recall are measured at the block level, edge level, and state level.

At the block level, the system is evaluated on whether it correctly estimates which blocks have been incorporated into the model, i.e., correctly detects vertex membership in each connected component [1]. The edge level evaluation measures whether the system correctly estimates which pairs of blocks are joined in the model, i.e., correctly identifies each edge present. The state level evaluation is the most precise, measuring if the system correctly estimates every block, edge, and edge label corresponding to relative block poses [1].

The results of the evaluation on the controlled dataset are presented in Table 1 of the paper [1]. The state accuracy is reported to be above 90% when parsing RGB data, with precision and recall nearly matching this performance. These results indicate that the system works well when the observed data match the expectations of the model. However, the performance is worse when parsing depth data, as the system has difficulty distinguishing between different states with the same adjacency structure. Nevertheless, the results on combined data show that including depth frames along with RGB does not perform worse than RGB alone [1].

In the evaluation on the child's play dataset, the system is tested on videos where state sequences are annotated, but keyframes are only annotated for a subset of the videos [1]. Leave-one-video-out cross-validation is used, and average metrics across folds are reported.

The results of the evaluation on the child's play dataset are presented in Table 2 of the paper [1]. These results show that the system performs reasonably well, although not as accurately as on the controlled dataset. The performance metrics indicate that the system is able to estimate block, edge, and edge label states with a certain level of accuracy, precision, and recall.

Conclusion

The paper presents a model for assembly processes and proposes a probabilistic inference algorithm [1]. The model is designed to parse block construction tasks and is evaluated on two datasets: a controlled dataset and a dataset collected from child behavioral experiments [1].

The system performs exceptionally well when the observed data align with the modeling assumptions [1]. The results show high accuracy, precision, and recall at the block, edge, and state levels [1]. This indicates that the system can accurately estimate which blocks have been incorporated into the model, identify pairs of joined blocks, and estimate block poses and relative positions [1].

Even under more challenging data conditions, the system still produces sensible results [1]. However, the performance is slightly worse when parsing depth data compared to RGB data, as the system struggles to distinguish between different states with the same adjacency structure [1]. Nevertheless, including depth frames along with RGB data does not result in worse performance, suggesting that the combination of modalities can be beneficial [1].

The evaluation on the child's play dataset, which represents a real-life setting, shows that the system performs reasonably well [1]. Although not as accurate as on the controlled dataset, the system is still able to estimate block, edge, and edge label states with a certain level of accuracy, precision, and recall [1].

The findings of the paper suggest new research directions, particularly in the areas of fine-grained action recognition and occlusion-robust computer vision [1]. The system's performance can be further improved by integrating it with an explicit action recognition system or by using IMU signals for improved template registration [1].

The paper also highlights the potential of exploring RGBD + IMU data fusion techniques using the multiple modalities available in the datasets [1]. IMU signals can be used as input for the action recognition system or to enhance the system's performance by deriving orientation estimates from the IMU signals [1].

In conclusion, the proposed model and inference algorithm demonstrate the ability to accurately parse block construction tasks. The system performs well under favorable data conditions and still produces reasonable results under more challenging conditions. The findings open up new research opportunities in action recognition, occlusion-robust computer vision, and data fusion techniques.