DeePoint: Visual Pointing Recognition and Direction Estimation

Shu Nakamura*   Yasutomo Kawanishi†   Shohei Nobuhara*   Ko Nishino*,†
*Graduate School of Informatics, Kyoto University   †RIKEN
https://vision.ist.i.kyoto-u.ac.jp/   https://grp.riken.jp/
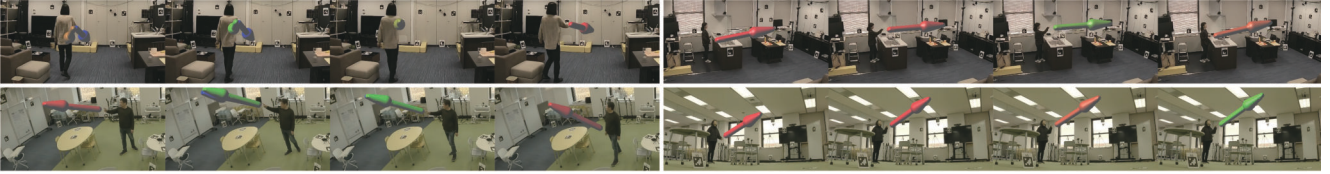
Figure 1. We introduce DeePoint, a neural pointing recognition and 3D direction estimator. DeePoint trained on our newly constructed DP Dataset recognizes when a person is pointing and estimates its 3D direction from video frames captured from a fixed-view camera. Each arrow depicts the pointing direction and its color is green when the person is pointing, red when not. DeePoint successfully recognizes when a pointing starts and ends and can estimate its 3D direction from the complex spatio-temporal coordination of the person's body.

Abstract

In this paper, we realize automatic visual recognition and direction estimation of pointing. We introduce the first neural pointing understanding method based on two key contributions. The first is the introduction of a first-of-its-kind large-scale dataset for pointing recognition and direction estimation, which we refer to as the DP Dataset. DP Dataset consists of more than 2 million frames of 33 people pointing in various styles annotated for each frame with pointing timings and 3D directions. The second is DeePoint, a novel deep network model for joint recognition and 3D direction estimation of pointing. DeePoint is a Transformer-based network which fully leverages the spatio-temporal coordination of the body parts, not just the hands. Through extensive experiments, we demonstrate the accuracy and efficiency of DeePoint. We believe DP Dataset and DeePoint will serve as a sound foundation for visual human intention understanding.

1. Introduction

Gauging a person's intent from passive visual observations is one of the key goals of computer vision research. Successful visual intent understanding would be essential for a wide range of applications including personal assistance, elderly care, and surveillance. Visual recognition of the gesticulations of a person is essential for this as they directly express those intents. Pointing, the act of extending one's (usually index) finger towards something in the person's view to call attention to it, is particularly important as it conveys explicit information about the person's interactions with the environment including conversations with others.

Despite the broad interest in gesture recognition, research on visual understanding of pointing has been surprisingly limited. Visual pointing interpretation requires both recognition (is the person pointing) and direction estimation (which direction is the person pointing). Past works have relied on special cameras, such as RGB-D sensors, or required the person to point in a specific way. For in-the-wild natural pointing understanding, we must be able to recognize and estimate their directions in 3D from regular RGB cameras. A typical scenario we consider is a person in a room pointing at various things around her while freely moving around, which is observed by cameras fixed to room corners.

Pointing recognition and direction estimation from fixed-view cameras is particularly challenging. The person is usually small in the view and the fingers can hardly be discerned. The hand can even be completely occluded by the person's body. The pointing gesture would also typically span only about half a second, which makes its recognition in the video hard. Estimating the direction becomes even more challenging. In a full HD video frame captured with a fixed corner camera in a typical living room, the index finger would span only about 30 pixels. Analytical modeling such as line regression to such observations would be futile. Even if that were possible, due to intra- and inter-personal variations of pointing, such estimations would be prone to error. Accounting for those variations would naturally necessitate a learning-based approach that directly regresses to

the intended directions. This is also, however, not straightforward, as the task is inherently spatio-temporal and, most important, large-scale data of pointing is difficult to collect and currently devoid in the community.

In this paper, as illustrated in Fig. 1, we make two key contributions to realize automatic visual recognition and direction estimation of pointing. The first is the introduction of a comprehensive dataset for pointing recognition and direction estimation. We refer to this as the DP Dataset. It consists of 2,800,000 frames of 33 people pointing at various directions in different styles captured in 2 different rooms. Each of these frames is annotated with whether the person is pointing or not, and, when pointing, the 3D direction intended by the pointing person. This first of its kind large-scale collection and annotation of natural pointing gestures is achieved semi-automatically with a combination of multi-view geometry and audio processing.

The second key contribution is DeePoint, a novel deep network model for joint recognition and 3D direction estimation of pointing. To overcome the challenges stemming from the fixed-view observations from a distance, our key idea is to leverage the whole-body appearance and motion to detect and estimate 3D pointing. For this, we introduce a Transformer model, inspired by the STLT [28], that fully leverages the spatio-temporal coordination of the body including the head and joints in addition to the hand. By incorporating the appearance of these as tokens and through cascaded attention transforms in space and time, we show that pointing gestures can be detected in time and their 3D directions can be estimated accurately.

We conduct extensive experiments to evaluate the effectiveness of DeePoint. We first evaluate the accuracy of recognition and direction estimation on the DP Dataset. We then evaluate the generalizability of DeePoint across different people and scenes. Through ablation studies, we also show that the spatio-temporal modeling of the body appearance and movements are essential for the task. We also conduct comparative studies with related works, including evaluation on the PKU-MMD dataset [5]. The experimental results collectively demonstrate the accuracy and efficiency of DeePoint. Our future work includes incorporating environmental cues and audio including spoken words to enhance the accuracy of pointing direction estimation, the challenge of which lies in realizing this without overfitting to the particular context. We believe DeePoint provides a sound foundation for these further studies.

2. Related Works

Gesture recognition has been a major topic of research in the computer vision community but research specific to pointing recognition and its direction estimation is fairly limited. We review works relevant to our approach of using the whole body for visual pointing understanding and

also on construction of large-scale real-world datasets for human behavior understanding.

2.1. Pointing Recognition

Early works of pointing recognition used wearable devices to measure pointing directions directly. Various devices such as magnetometers [2] and IMUs [3] have been adopted. Since the person to be measured must wear a dedicated device for pointing, however, the applications of these methods were limited.

Most past pointing recognition methods require special camera setups. These works include those that require multiple cameras [15, 34, 6, 14, 20, 25], RGB-D cameras [31, 13, 1, 10], or depth sensors [7, 8]. From the visual observations captured with these specialized cameras or setups, these methods estimate pointing direction in mainly two ways: geometry-based or learning-based.

Geometry-based approaches first locate 3D coordinates of specific body parts, *e.g.*, face, hand, or fingertip, and calculate the direction of pointing by extending the line connecting them. Results of such methods can be very noisy as the detection and triangulation of these body parts can be unreliable. Learning-based approaches estimate the 3D direction from the observed appearance of the body parts, *e.g.*, hands or arms. Both of these approaches can achieve accurate pointing direction recognition when certain imaging conditions are met, *e.g.*, the person is in fairly near distance from the camera and showing a perfect side-view of the pointing. They, however, fundamentally rely on multi-view observations or direct depth perception, which preclude their use with regular RGB cameras.

A few recent works achieve pointing recognition from a single RGB image to alleviate the needs for special camera settings. Estimating pointing direction in 3D is inherently challenging, as 3D locations of body parts or detailed appearance of hands cannot be captured by a normal camera. Past works resolve this by limiting the allowed postures of the target person, for instance, by requiring the person to stand upright with her arm fully extended when pointing [4, 30]. Jaiswal et al. [19] introduced a ConvNet pointing direction estimation, but is limited to when the person is standing in front the camera with her body facing towards it. These methods, in essence, recognize a special pre-defined body posture as pointing, which does not generalize across people and scenes. Our DeePoint, in contrast, realizes automatic visual recognition and direction estimation of pointing by a person freely moving, regardless of walking or sitting, in a room-size area from a single view of a regular camera. To our knowledge, this is the first work to achieve 3D pointing understanding in the wild.

2.2. Action Recognition

Pointing can be viewed as a special gesture or action. General gesture and action recognition research has a long history in computer vision. Many benchmark datasets have been released, such as UCF101 [22] and ActivityNet [12]. As spatio-temporal visual information becomes crucial for recognizing actions, a variety of approaches for capturing temporal relations of body and other contextual movements have been proposed such as CNN+LSTM [11] and 3DRes-Net [16].

More recently, Transformers [32] have been applied to learn such spatio-temporal coordination through their attention mechanism [33, 35]. Among them, Radevski et al. [28] proposed a two-stage transformer model which captures spatial relationships of object with the first transformer for each frame and temporal relationships of their movements with the second transformer. We build on this idea of decoupling spatial and temporal information aggregation with two cascaded Transformer encoders and extend it to encoding body postures and their temporal coordination to achieve accurate pointing recognition and direction estimation.

2.3. 3D Direction Annotation

For learning-based 3D direction estimation, annotation of images with 3D vectors becomes essential. This task is, however, extremely challenging, if not impossible, to achieve manually as the annotator needs to somehow indicate the projected 2D direction from a 3D ground truth in mind on the 2D image plane. Past works have mitigated this difficulty by exploring automatic means to directly obtain the 3D ground truth. Das [7, 8] attached a colored marker or an IMU to the index finger to obtain ground-truth pointing directions. This is possible for their method as they rely on direct depth perception for pointing recognition and artificial appearance of the person does not affect the input.

Other methods leverage multi-view geometry of cameras to compute 3D directions of 3D gaze and pointing. Kellnhofer *et al*. [21] proposed Gaze360, a large-scale 3D gaze tracking dataset. The data was collected with an omnidirectional camera that simultaneously captures subjects and their gaze targets. By using an AR marker as the gaze target, the authors realize automatic annotation of the 3D location of the target. Nonaka *et al*. [26] introduced GAFA, a 3D gaze dataset with per-frame 3D gaze annotations. The gaze directions were captured with an eyeglass gaze tracker. For the ground truth head and body orientations, they used body- and head-mounted cameras and AR markers attached to compute the 3D orientations via SLAM. We automatically annotate our DP Dataset with accurate 3D pointing directions by identifying the pointed AR markers in the scene from audio and by computing the 3D directions to them with multi-view geometry. We also obtain the pointing tim-
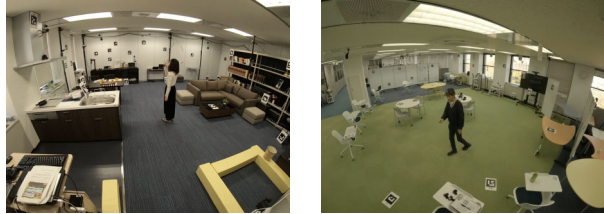
(i) Living Room   (ii) Office

Figure 2. The two environments of DP Dataset. The example frames are captured by the cameras outlined in red in Fig. 3.

ing and duration with synchronized audio. We believe this multi-modal automatic annotation would be useful in other dataset annotation tasks.

3. DP Dataset

Our first key contribution is the first-of-its-kind large-scale dataset for pointing recognition and 3D direction estimation. We make this dataset and code available to the public[1].

3.1. Dataset Capture

A large-scale dataset of videos capturing people pointing in various directions as they naturally roam around and sit and stand in an environment with accurate timing and 3D direction annotations is essential for exploring learning-based approaches to visual pointing understanding. The dataset desiderata include variations in the people spanning age and gender, the viewpoint and viewing directions, the pointing styles and timings including duration, the pointed directions, the behaviors of people such as standing, walking, and sitting, and the overall environments in which the people are immersed. Also, in order to use the natural appearance of people, they should not wear specific measurement devices that affect their appearance, such as motion capture devices, special markers, or gaze measurement devices, as a learning-based approach would overfit to them. To the best of our knowledge, there are no large-scale public datasets for pointing recognition and direction estimation that fulfill these.

We introduce *DP Dataset*, a first-of-its-kind large-scale pointing dataset, which consists of 2,800,000 frames of 33 people of various ages and different genders pointing in a wide range of directions in different styles in two different rooms captured from a variety of viewing directions with multiple fixed-view cameras at room-scale distances. Most important, the dataset includes annotations of pointing timings and their 3D directions for each and every frame.

As shown in Fig. 2, we constructed a data capture imaging setup for two different rooms. One is a living room with a kitchen and sofa, and another is an open office with chairs,

---

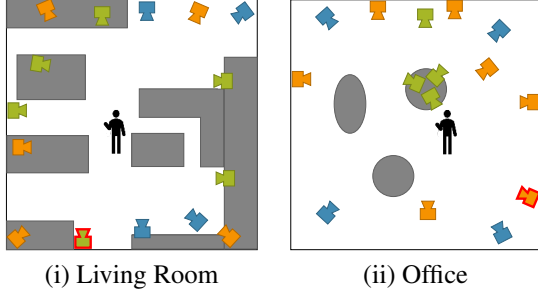(i) Living Room   (ii) Office

Figure 3. Camera layout of DP Dataset. We mount cameras at fixed viewpoints in the room to capture the pointing gestures from a variety of directions at once. The orange cameras are installed on the ceiling, the blue ones are put on the floor, and the green ones are installed on the mid-level. The gray objects depict tables, sofas, and obstacles.

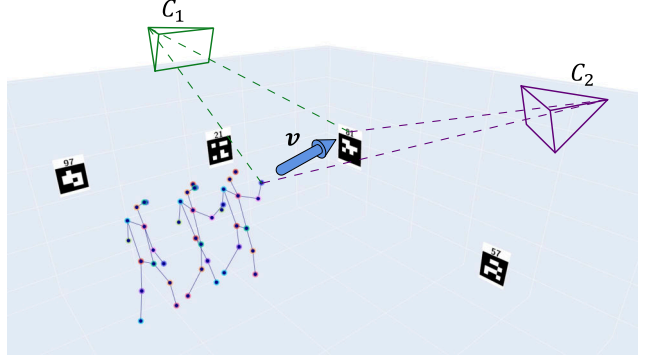Figure 4. Example frames from the DP Dataset (people are cropped).

Figure 5. By identifying the marker to which the person is pointing from recorded audio and triangulating the hand location, we compute the unit 3D vector annotation for the 3D pointing direction.

desks, and whiteboard, which we refer to as Living Room and Office, respectively. Both rooms are about $64\,\mathrm{m}^2$. As depicted in Fig. 3, we installed 15 GoPro cameras in each room to capture people in them and calibrated all the cameras so that we could triangulate the 3D position of each joint and marker in the environment. They were installed in various locations in the room pointing towards the center so that a person in the room can be captured from all directions roughly uniformly. For this, we mounted the cameras on the tables, walls, the floor, and the ceiling. All cameras were synchronized at the beginning of the capture.

We captured videos in 2.7K resolution at 60fps and drop the frame rate to 15fps for the dataset we use for our experiments. The raw dataset can also be released upon request. To annotate the pointing direction, we installed roughly 40 ArUco [29] markers randomly on tables, walls, the floor, and the ceiling, in each room. Each marker is observed by multiple cameras and its 3D location is recovered with triangulation.

As shown in Fig. 4, pointing style varies from person to person and the dataset should capture this variation as much as possible. For the dataset, we collected a total of 33 male and female participants, uniformly ranging in generations from their twenties to sixties. We captured each participant separately for about 5 minutes in each room. Each participant was free to walk around the room and point to markers freely selected by themselves with their dominant hand, but asked to verbalize the marker ID and click and hold down on a handheld wireless mouse when pointing. They were also allowed to point to the markers while sitting on a chair or sofa. They pointed to a marker once every 3 to 5 seconds while moving in the room. For each session, we collected 15 videos from the different fixed-view cameras.

3.2. Pointing Timing and 3D Direction Annotation

We fully annotate DP Dataset with pointing timings, *i.e.*, the start and end of a pointing instance, and the 3D directions for all pointing instances. We annotate the pointing timings by asking the participants to indicate the start and duration of when he or she points to a marker. This is achieved by providing the participants with a small click button, for which we simply used a tiny wireless mouse, held in the non-dominant hand so that it is not visible from the camera. Participants pressed the button when they started pointing to a marker and held it down until their pointing gesture finished. The duration is typically less than a second for a natural pointing behavior.

As depicted in Fig. 5, we automatically annotate the 3D directions of each pointing instance with multi-view geometry. Participants were asked to verbally express which marker they were pointing to, whose voice was recorded by the observing cameras. By manually identifying the marker ID from the recorded voice, we know the 3D coordinates of the target pointing direction. To recover the other end of the 3D vector, *i.e.*, from where that marker is pointed, we first apply 2D pose estimation to the videos captured by the cameras and calculate the 3D hand locations based on triangulation using only high-confident 2D pose estimation results. We use OpenPifPaf [24] as the pose estimator, but any method that is sufficiently accurate can replace it. Ac-
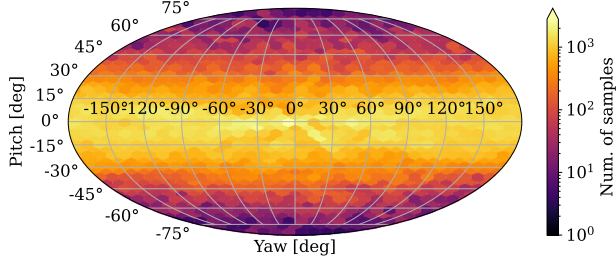
Figure 6. The 3D angular distribution of pointing directions in the DP Dataset shown with Mollweide projection. A variety of pointing behaviors with a wide range of pitch and yaw are captured in the dataset.
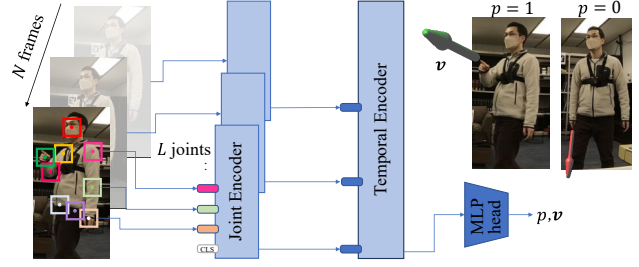

Figure 7. DeePoint consists of two Transformer encoders which we refer to as Joint Encoder and Temporal Encoder. Joint Encoder learns to model the spatial coordination of body parts and Temporal Encoder learns to extract their temporal coordination to jointly recognize and estimate the 3D direction of pointing from RGB video frames.

curate pointing directions were calculated from each pair of a 3D hand location and the pointed 3D marker location. We describe each pointing direction as a 3D unit vector.

In total, the dataset contains about 2,800,000 frames of 33 people. Frames with pointing span 770,000 frames capturing 6355 unique pointing instances. Although the number of located ArUco markers in a room was limited to about 40, we were able to collect a large variety of pointing directions in the dataset as the participants were allowed to move around and change their postures freely in the rooms. Figure 6 shows the 3D angular distribution of pointing directions. The histogram clearly shows that we were able to capture a wide variety of pointings in the dataset. Note that each of these instances are captured with a wide range of viewing directions using the 15 cameras.

4. DeePoint

We introduce DeePoint, a novel method for accurate pointing recognition and 3D direction estimation. Unlike past works, the method does not rely on specific poses taken by the target person and only requires regular RGB video frames as input. As depicted in Fig. 7, DeePoint is a Transformer-based model which leverages attention for spatio-temporal information aggregation as first introduced for video understanding by Radevski *et al.* [28]. In contrast to learning the spatio-temporal coordination of objects in a scene for video understanding, we leverage the STLT architecture [28] to learn the structured spatio-temporal coordination of body parts of a person when she is pointing and simultaneously detect and estimate its 3D direction. Given a sequence of input frames, DeePoint first detects the joints using an off-the-shelf 2D human pose estimator [24], and extracts visual features around them. The visual features are first processed by Joint Encoder in a frame-wise manner, and then fed to Temporal Encoder to integrate features from multiple frames. The output of Temporal Encoder is transformed by an MLP head to the probability p indicating whether the target is in a pointing action, and its 3D direction ν in the camera coordinate system.

Pose Feature Extraction We use the output of the third block of ResNet-34 [18] pre-trained by ImageNet-1K [9] as the backbone for extracting 256-channel visual features from each of the input video frames, and apply ROI align [17] around each joint to obtain $3 \times 3 \times 256$ feature vectors of a constant size regardless of the apparent joint size. These feature vectors are then projected by a learnable linear layer to 192 dimensions.

Though these visual features around the joints collectively cover a certain area over the target person, they do not explicitly describe the relative positions between the joints, *i.e.*, the pose. We encode the 2D pose information by two additional features per joint: the joint indices and the 2D relative position w.r.t. the midpoint of the shoulders normalized by the bounding box size. Both the joint indices and the normalized relative positions are projected to the size of the visual features so that they are added as positional encodings [32]. These projections are randomly initialized and refined in the training.

For undetected joints or joints with confidence below a certain threshold due to, for example, occlusions, corresponding visual features are padded with a dummy tensor and ignored in the subsequent steps by masked attention [32].

Joint Encoder The Joint Encoder takes the visual features of human joints and a class token, and computes multi-head attention [32] between them. We set Joint Encoder to accept $L = 17$ tokens corresponding to the joints detected by pose estimation. Joint Encoder processes such tokens with 6 iterations of attention layers, and returns the output corresponding to the class token for the last attention layer as the final output.

Temporal Encoder Our Temporal Encoder takes the output of Joint Encoder of the current frame together with those of past N frames as input tokens. Temporal Encoder has 6

layers of multi-head attention, and the output token corresponding to the current frame at the last layer is used as the output of Temporal Encoder.

MLP head The output of Temporal Encoder is transformed by an MLP into the pointing probability p and the pointing direction ν. The pointing probability p is implemented as binary classification and the MLP outputs a 2-dimensional vector normalized by the sigmoid function. The pointing direction ν is first regressed as a 3-dimensional vector of arbitrary norm, and then normalized to be a unit vector.

Training We train DeePoint using DP Dataset in a supervised manner, by measuring the cross entropy of p and the angular error of ν between their ground truths. The weighting parameter to balance these two terms is determined empirically. During training, we randomly sampled frames so that pointing and non-pointing frames appear evenly.

5. Experimental Results

Network architecture DeePoint uses visual features around the detected joints to encode the body posture and its specific instantiation. In addition, we may also leverage visual features of the whole body and even encode the entire captured image. Since each token of Joint Encoder is a 192-dimensional vector, we can add these contextual visual features that encode the body and scene appearance into the class token since it is not associated with a specific joint.

As the same for the visual features at each joint, we can apply the same pre-trained ResNet-34 to the image cropped by the bounding box of the detected person and the entire image, apply ROI align to obtain $16 \times 16 \times 256$ feature vectors and project them into 192-dimensional vectors with the same linear projection. These vectors are then added to the learnable class token. In what follows, we denote the barebone DeePoint as *DP*, a variant adding the whole-body visual feature to the class token as *DP-B*, and yet another variant adding both the whole-body and the entire-image visual features to the class token as *DP-BI*.

Data split Our DP Dataset consists of roughly 2,800,000 frames of captured sessions of 33 subjects in two different rooms (Living Room and Office). We define the following three splits for evaluation.

Split-T (temporal split) Each session of the subjects is split into 70%, 15%, and 15% from the beginning and used in the training, validation, and test sets, respectively. The three sets share the same subjects and the scenes, but not the pointing instances and their directions. This split lets us evaluate the intra-personal accuracy of DeePoint.

Model	*Split-T*	*Split-S*	*Split-P*
DP	14.05	**17.52**	**13.85**
DP-B	**13.66**	17.63	13.93
DP-BI	14.12	17.62	14.91

Table 1. Pointing direction estimation errors, denoted in degree. We can observe that the proposed model generalizes well in every split.

Model	*Split-T*	*Split-S*	*Split-P*
DP	0.625/0.838	0.629/0.685	**0.476**/0.816
DP-B	0.627/**0.852**	0.597/0.732	0.445/0.823
DP-BI	**0.650**/0.837	**0.634**/**0.740**	0.456/**0.855**

Table 2. Recall (left) and precision (right) for the pointing action detection. Note that these values are calculated frame by frame and the percentage of pointing actions that are missed completely are much lower.

Split-S (scene split) The training set does not share the same room with the validation and test set. That is, the training set is only taken from Living Room, and the validation and the test sets are from Office. This split lets us study the cross-scene accuracy of DeePoint.

Split-P (person split) Each of the 33 subjects appears only in one of the training, validation, or test set. We allocated 25, 4, and 4 subjects for the training, validation, and test sets, respectively. This split lets us evaluate the inter-personal accuracy of DeePoint.

Tables 1 and 2 each reports the mean angular errors of pointing direction and recall/precision of pointing detection by frame. Each model is trained using the training set with learning rate = 10^{-4} with Adam [23] optimizer and batch size = 64 until convergence. We use the best parameter within the epochs in terms of angular error measured with the validation set.

The results provide insights into the role of the body and scene context. For the intra-personal split (*Split-T*), *DP-B* and *DP-BI* detects pointing better than *DP* as they can leverage the access to scene context. Performance for direction estimation and precision gets worse for *Split-S*, which indicates changes in the way of pointing and background could affect the performance and the importance of training with a dataset that contains multiple venues. On the other hand, as for *Split-P*, recall is relatively low, which indicates the way of pointing differ in people. While integrating whole-body and scene context contributes to improving the performance for 3D direction estimation, the performance for pointing detection is better without them for *Split-S* and *Split-P*, likely due to overfitting. How to encode personal (*i.e.*, body appearance) and scene (*i.e.*, image) context in DeePoint such that we may fully leverage their representational power while avoiding overfitting is a challenge we

Figure 8. Pointing direction estimation by DeePoint trained with *Split-T* (*i.e.*, *DP*). In each image, the blue arrow denotes the ground truth direction and the other arrow denotes the estimated 3D direction by DeePoint. The color of the prediction arrow represents the result of pointing action recognition. It is green when the person is pointing ($p = 1$), red when not ($p = 0$), and gradually transitions between the two colors based on estimated probability. Note how DeePoint correctly recognizes the timing of pointing. For instance, it learns to recognize when the person looks away as the finish of pointing and finds the onset of pointing from change in speed of the movements of the body coordination.

| ←60px→ | ←98px→ | ←112px→ | ←113px→ |

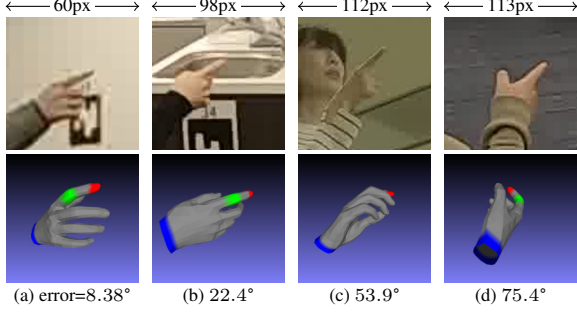(a) error=8.38° (b) 22.4° (c) 53.9° (d) 75.4°

Figure 9. Examples of HandOccNet reconstruction on the DP Dataset, sorted by index finger direction error. HandOccNet fails to reconstruct the pointing index finger for most cases, especially as the hands are small in regular videos of people, showing the fragility of 3D hand reconstruction-based pointing understanding.
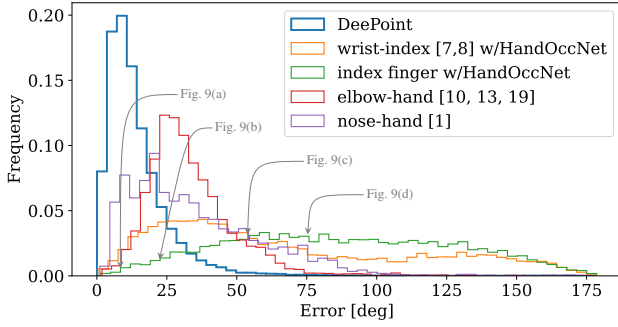

Figure 10. Error histograms of past methods compared with that of DeePoint evaluated on the test data of the DP Dataset. DeePoint clearly outperforms all.

will explore in future work.

As can be seen in Fig. 8, in general, DeePoint achieves high recall and reasonable angular accuracy, especially for a completely passive method relying only on viewpoint from relatively far distance. Even though roughly 18 degrees of error may appear large, given the relative distance to the objects in the scene, in many cases, it is sufficient to identify what is actually pointed at. Note that the detection recall and precision are calculated by each frame and only a fraction of pointing actions are missed completely. For example, *DP* on *Split-T* test set missed only 5.8% of pointing actions (94.2% recall).

5.1. Ablation Study On Temporal Window

The size of the temporal window, *i.e.*, the number of tokens N given to Temporal Encoder is set to $N = 15$ as the default value. Table 3 shows the results with different values of N from $N = 1$ to $N = 30$. $N = 1$ corresponds to single-shot pointing detection and direction estimation, and $N = 5$, 15, and 30 correspond to 1/3, 1, and 2 seconds of the observed video.

From these results, we can conclude that $N = 15$, which is used in DeePoint, is a reasonable design choice as the

Temporal window	Angular error ($\downarrow$)	Prec./Rec. ($\uparrow$)
$N = 1$	17.08°	0.519/0.801
$N = 5$	14.90°	0.585/0.828
$N = 15$	14.05°	0.625/**0.838**
$N = 30$	**13.58°**	**0.637**/0.833

Table 3. Contributions of the size of the temporal window N. We can observe that $N = 15$ corresponding to 1 second of the observation is a reasonable design choice as the performance gain by $N = 30$ is marginal.

performance gain by $N = 30$ is marginal, while $N = 1$ and $N = 5$ do not perform well, especially in action detection. This result can be interpreted intuitively that most pointing instances can last up to a second and not shorter than 1/3 seconds, and $N = 15$ is a reasonable length to cover such actions.

5.2. Comparison with Baseline Methods

We implement baseline methods that represent past methods and evaluate them using the test split of the DP dataset, and compare their results against that of DeePoint. Directly evaluating DeePoint on the datasets used in the past methods is not possible, as most of them are simply not published [1, 4, 10, 13, 19, 30]. Even when they are, they contain only images (not videos) and capture only hands or arms [8, 31]. To the best of our knowledge, the only exception is PKU-MMD [5], a video dataset annotated with various action timings, including pointing. We'll discuss it in Sec. 5.3.

To evaluate the accuracy of a single-view learning-based approach, we use HandOccNet [27] to recover a 3D hand mesh from a single image and use the recovered hand to estimate the pointing direction. Since pointing is a manual gesture, it may appear possible to estimate pointing direction by connecting vertices of the mesh. As shown in Fig. 9, we applied HandOccNet to hand image regions extracted from the DP dataset to test this. We tried two alternatives for direction estimation: from the wrist to the tip of the index finger (*i.e.*, from the center of the blue vertices to that of the red ones in Fig. 9) [7, 8] and from the base to the tip of the index finger (from green to red). Figure 10 shows that the results are poor. This is because, as can be seen in Fig. 9, HandOccNet fails to reconstruct the index finger accurately for most cases due to the low resolution of the hand regions.

We also replicated geometry-based approaches using the 3D keypoints of the DP dataset. Most of these methods calculate the pointing direction by estimating the 3D coordinates of keypoints and connecting them (elbow to wrist [10, 13, 19] or face to hand [1]). The triangulated keypoints in the DP Dataset can be used to simulate these approaches (elbow to hand or nose to hand). The results are also depicted in Fig. 10 and they clearly show that our

Figure 11. Qualitative evaluation with the PKU-MMD dataset [5]. Note that our model is not retrained on the PKU-MMD and applied out-of-the-box. DeePoint generalizes well to a completely different dataset.
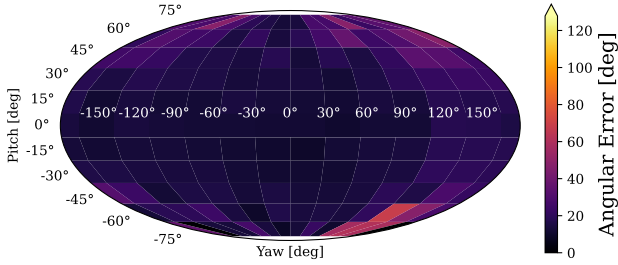

Figure 12. Mean angular error (*i.e.*, 3D direction estimate accuracy) distribution for each ground truth pointing direction. The results show that errors increase for pointing with high/low pitches.

DeePoint estimates are more accurate by a wide margin.

These results clearly show that modeling the whole body movements is essential to achieve accurate pointing direction estimation, especially for in-the-wild scenarios in which the person is captured from afar.

5.3. Cross-Validation with PKU-MMD dataset

Although there does not exist a large-scale pointing dataset in the community, PKU-MMD [5], a video dataset of various actions, contains a small number of pointing videos in it. We used PKU-MMD Phase # 2 which originally consists of 7,000 action instances of 41 action classes, performed by 13 subjects. We use one of the 41 action classes named "pointing to something with finger" (class 24) for validation. The dataset contains 817,314 non-action frames, 497,296 non-pointing action frames, and 6,708 pointing frames. The pointing directions, however, are not annotated in PKU-MMD and we can only conduct qualitative evaluations.

Figure 11 shows the pointing directions estimated by DeePoint (*DP*) trained with *Split-T* of DP Dataset. Note that the model is not fine-tuned with PKU-MMD (as there are no ground truth directions to fine-tune on). The detection accuracy was 72.2% for the pointing + non-action frames, and 69.9% for the entire pointing + non-pointing + non-action frames. From these results, we can conclude that our DeePoint trained with DP Dataset generalizes reasonably well to new scenes and subjects.

5.4. 3D Direction Accuracy Across Pointing Directions

To better understand the 3D direction estimation accuracy of DeePoint, we evaluate the relationship between the ground truth pointing directions and the angular error of 3D direction estimates. Figure 12 shows the angular error as a distribution over ground-truth pointing directions in Mollweide projection. The results show that DeePoint struggles with pointing with high yaw ($> 120°$) with high/low pitches ($> 60°$ or $< -60°$). The error is especially high with low pitches, where the person points down while facing away, which means the arms are often occluded by the body.

6. Conclusion

In this paper, we introduced a novel method for pointing recognition and 3D direction estimation. DeePoint leverages the spatio-temporal coordination of a person's body to recognize and estimate the timing and direction of pointing from video frames captured from a fixed-view in a relatively far distance. We also introduced the DP Dataset, the first large-scale visual pointing dataset with full annotation of the timings and 3D directions of natural pointing behaviors of a variety of people in different scene contexts. We believe these two fundamental contributions significantly advance visual pointing understanding and serve as a sound foundation for human behavior and intent understanding. We make all the data and code publicly available to catalyze further advances in this field.

Limitation DeePoint can incorporate scene context but only as 2D images from the fixed viewpoint. Our future work includes incorporating such explicit visual cues of the environment, *e.g.*, object detection in the scene to aid in narrowing down the exact object the person is pointing to. We also plan to explore the use of audio, particularly spoken words for this. Incorporating more scene context in these forms has the danger of overfitting to the particular context. We believe DeePoint provides a robust springboard for these further studies.

References

[1] Bita Azari, Angelica Lim, and Richard Vaughan. Commodifying pointing in HRI: Simple and fast pointing gesture detection from RGB-D images. In *Proc. of International Conference on Computer and Robot Vision*, pages 174–180, May 2019. 2, 8

[2] Richard A. Bolt. "Put-That-There": Voice and gesture at the graphics interface. In *Proc. of ACM SIGGRAPH*, pages 262–270, July 1980. 2

[3] Denis Broggini, Boris Gromov, Alessandro Giusti, and Luca Maria Gambardella. Learning to detect pointing gestures from wearable IMUs. In *Proc. of AAAI Conference on Artificial Intelligence*, pages 8051–8052, Feb. 2018. 2

[4] Zuzana Černeková, Nikos Nikolaidis, and Ioannis Pitas. Single camera pointing gesture recognition using spatial features and support vector machines. In *Proc. of European Signal Processing Conference*, pages 130–134, Aug. 2007. 2, 8

[5] Liu Chunhui, Hu Yueyu, Li Yanghao, Song Sijie, and Liu Jiaying. PKU-MMD: A large scale benchmark for continuous multi-modal human action understanding. *arXiv:1703.07475*, Mar. 2017. 2, 8, 9

[6] Roberto Cipolla, Paul A. Hadfield, and Nicholas J. Hollinghurst. Uncalibrated stereo vision with pointing for a man–machine interface. In *Proc. of IAPR Workshop on Machine Vision Applications*, pages 163–166, 1994. 2

[7] Shome S. Das. Precise pointing direction estimation using depth data. In *Proc. of International Symposium on Robot and Human Interactive Communication*, pages 202–207, Aug. 2018. 2, 3, 8

[8] Shome S. Das. A data-set and a method for pointing direction estimation from depth images for human-robot interaction and VR applications. In *Proc. of International Conference on Robotics and Automation*, pages 11485–11491, May 2021. 2, 3, 8

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009. 5

[10] Naina Dhingra, Eugenio Valli, and Andreas Kunz. Recognition and localisation of pointing gestures using a RGB-D camera. In Constantine Stephanidis and Margherita Antona, editors, *Proc. of HCI International 2020 - Posters*, pages 205–212, July 2020. 2, 8

[11] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, June 2015. 3

[12] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, June 2015. 3

[13] Ana Fernández, Luca Bergesio, Ana M. Bernardos, Juan A. Besada, and José R. Casar. A kinect-based system to enable interaction by pointing in smart spaces. In *Proc. of IEEE Sensors Applications Symposium*, Sept. 2015. 2, 8

[14] Dai Fujita, Takashi Komuro, Michael M. Bronstein, and Carsten Rother. Three-dimensional hand pointing recognition using two cameras by interpolation and integration of classification scores. In *Proc. of European Conference on Computer Vision Workshops*, pages 713–726, Sept. 2015. 2

[15] Masaaki Fukumoto, Kenji Mase, and Yasuhito Suenaga. Real-time detection of pointing actions for a glove-free interface. In *ProcİAPR Workshop on Machine Vision Applications*, pages 473–476, Dec. 1992. 2

[16] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3D residual networks for action recognition. In *Proc. of IEEE/CVF Internattional Conference on Comptuer Vision Workshops*, pages 3154–3160, Oct. 2017. 3

[17] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *Proc. of IEEE Internattional Conference on Comptuer Vision*, pages 2961–2969, Oct. 2017. 5

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, June 2016. 5

[19] Shruti Jaiswal, Pratyush Mishra, and G.C. Nandi. Deep learning based command pointing direction estimation using a single RGB camera. In *IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering*, pages 1–6, Nov. 2018. 2, 8

[20] Roland Kehl and Luc Van Gool. Real-time pointing gesture recognition for an immersive environment. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 577–582, Jan. 2004. 2

[21] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. Gaze360: Physically unconstrained gaze estimation in the wild. In *Proc. of IEEE/CVF Internattional Conference on Comptuer Vision*, pages 6912–6921, Oct. 2019. 3

[22] Mubarak Shah Khurram Soomro, Amir Roshan Zamir. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, Dec. 2012. 3

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 6

[24] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. OpenPifPaf: Composite fields for semantic keypoint detection and spatio-temporal association. *IEEE Transactions on Intelligent Transportation Systems*, 22(8):13498–13511, Mar. 2021. 4, 5

[25] Kai Nickel and Rainer Stiefelhagen. Pointing gesture recognition based on 3D-tracking of face, hands and head orientation. In *Proc. of International Conference on Multimodal Interfaces*, pages 140–146, Nov. 2003. 2

[26] Soma Nonaka, Shohei Nobuhara, and Ko Nishino. Dynamic 3D Gaze From Afar: Deep gaze estimation from temporal

eye-head-body coordination. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2192–2201, June 2022. 3

[27] JoonKyu Park, Yeonguk Oh, Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Handoccnet: Occlusion-robust 3d hand mesh estimation network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 8

[28] Gorjan Radevski, Marie-Francine Moens, and Tinne Tuytelaars. Revisiting spatio-temporal layouts for compositional action recognition. In *Proc. of British Machine Vision Conference*, Nov. 2021. 2, 3, 5

[29] Francisco J. Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and Vision Computing*, 76:38–47, Aug. 2018. 4

[30] Yuuichiro Shiratori and Kazunori Onoguchi. Detection of pointing position by omnidirectional camera. In *Proc. of International Conference on Intelligent Computing: Intelligent Computing Theories and Application*, pages 774–785, Aug. 2021. 2, 8

[31] Dadhichi Shukla, Özgür Erkent, and Justus Piater. Probabilistic detection of pointing directions for human-robot interaction. In *Proc. of International Conference on Digital Image Computing: Techniques and Applications*, pages 1–8, Nov. 2015. 2, 8

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. of Annual Conference on Neural Information Processing Systems*, pages 6000–6010, Dec. 2017. 3, 5

[33] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Zhengrong Zuo, Changxin Gao, and Nong Sang. OadTR: Online action detection with transformers. In *Proc. of IEEE/CVF Internattional Conference on Comptuer Vision*, pages 7565–7575, Oct. 2021. 3

[34] Hiroki Watanabe, Hitoshi Hongo, Mamoru Yasumoto, and Kazuhiko Yamamoto. Detection and estimation of omni-directional pointing gestures using multiple cameras. In *Proc. of IAPR Workshop on Machine Vision Applications*, pages 345–348, Jan. 2000. 2

[35] Jiewen Yang, Xingbo Dong, Liujun Liu, Chao Zhang, Jiajun Shen, and Dahai Yu. Recurring the transformer for video action recognition. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14063–14073, June 2022. 3