IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 November 2023, accepted 15 November 2023, date of publication 7 December 2023,
date of current version 14 December 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3340916*

RESEARCH ARTICLE

# TOPSIS Aided Object Pose Tracking on RGB Images

**MATEUSZ MAJCHER AND BOGDAN KWOLEK**
Department of Computer Science, AGH University of Science and Technology, 30-059 Kraków, Poland
Corresponding author: Bogdan Kwolek (bkw@agh.edu.pl)

This work was supported by the Polish National Science Center (NCN) under a Research Grant 2017/27/B/ST6/01743.

**ABSTRACT** In the problem of object pose estimation, one way to cope with the effect of ambiguity is to use multiple hypotheses. In this work, rather than generating the output pose based on a single object pose, our objective is to enable the system to be aware of the potential object ambiguity through maintaining multiple pose hypotheses. Firstly, we propose a pipeline for 6D object pose tracking on RGB images, wherein a key design is a fuzzy TOPSIS module that takes full advantage of multi-criteria decision making under uncertainties. Secondly, using decision variables determined on features that are frequently utilized in object pose estimation or tracking like segmented masks, fiducial keypoints, and distance transform the proposed method permits achieving tangible performance gains. An hourglass-based neural network is proposed to jointly detect object keypoints, predict the object's non-occluded part, and to predict the object's occluded part. To verify our designs, we conducted thorough experiments on the YCB-Video benchmark dataset. Besides, our method achieves competitive results in terms of ADD scores on the YCB-Video, showing that maintaining multiple pose hypotheses is beneficial to the task of object pose tracking. We observe that our method achieves competitive results against six recent methods estimating object pose from single frame and two SOTA object pose trackers. Extensive ablation studies verify our design choices.

**INDEX TERMS** Object pose tracking, multi-criteria decision making, multi-task neural networks, handling uncertain data, fuzzy TOPSIS.

## I. INTRODUCTION

The aim of 6 DoF (Degrees of Freedom) object pose estimation is to infer the object's 3D orientation and 3D translation with respect to the camera. Due to lighting changes and occlusions, accurately estimating object pose from a single RGB image is a challenging task. Although several recent methods can achieve object pose recovery with high accuracies on RGB images [26], the ill-posedness still makes this task very challenging. Because of the potential applications of 6D pose estimation from single-image several works considered this problem from an applications perspective, [19], [34]. Such single-image approaches re-estimate poses from scratch for every new frame [27], [33], [36]. Owing to leveraging temporal information from former frames the methods called object pose trackers aside from improving the accuracy of pose recovery additionally permit delivering smooth 6D trajectories.

Due to rapid progress in deep learning in the last decade, the methods for 6D object pose estimation achieve promising results on benchmark datasets recorded in controlled scenarios [10]. Such data-driven methods estimate 6D object pose by direct pose regression [39] or indirectly by estimating object keypoints and then executing a PnP (Perspective n-Point) algorithm. Regression-based techniques usually employ CNNs (Convolutional Neural Networks) to learn a mapping from the training images to ground-truth 6D object poses. Seminal work on end-to-end pose estimation resulted in a complex architecture called PoseCNN [34], which tackles the pose estimation problem through three related tasks: semantic labeling, recovery of translation from the estimated 2D object center, and inference of the rotation. Indirect approaches generally calculate sparse or dense 2D-3D correspondences and afterwards use a PnP

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Zhou.

algorithm to recover the 6D object pose. Such approaches detect some pre-defined semantic keypoints [12], [23], [24], [25] or alternatively estimate the pose through learning a model for predicting the corresponding 3D coordinates for object pixels [17]. A recently proposed approach [38] utilizes a graph convolutional network (GCN) to transfer the domain-invariant keypoints geometry from synthetic images to real ones. The algorithm predicts the locations of eight object keypoints and calculates the 6D pose using the PnP algorithm. A disadvantage of keypoint-based methods is their sensitivity to occlusions. Given 2D-3D correspondences, the 6D pose is usually recovered using the PnP algorithm in combination with the RANSAC-based (Random Sample Consensus) scheme.

Most of the existing techniques for 6D object pose estimation and object pose tracking are single hypothesis methods [10]. There are only a few notable works that leveraged a multiple hypotheses approach. In [6] a Rao-Blackwellized particle filter, called PoseRBPF, for 6D object pose tracking has been proposed. PoseRBPF estimates the 3D translation of object undergoing tracking along with a distribution over the 3D rotation. In real-world vision systems for object pose tracking, measurement ambiguity can arise as objects may have symmetrical shapes and undergo occlusions, making it hard or impossible to determine a single consistent object pose estimate. One way to cope with the effect of ambiguity is to calculate multiple hypotheses in each frame such a consistent set of hypotheses could be covered. In this work, rather than generating the output pose based on a single object pose at a time, as most existing systems do [10], our objective is to enable the system to be aware of the potential object ambiguity through maintaining multiple pose hypotheses.

Although some noteworthy segmentation-driven methods for object pose estimation exist [12], to the best of our knowledge, no significant method that jointly predicts an object's non-occluded part and object's occluded part has been done until now. The visibility of the object is an important factor in the tracking, particularly on images from a single RGB camera [35]. In [22] a GAN (Generative Adversarial Network) has been utilized to recover the occlusion part of the object. Although some work utilized both object segmentation and keypoint detection [25], the number of papers focusing on using object segmentation and object keypoints for multi-criteria decision making is very limited. Using both keypoints and segmentation in object pose tracking with the support of a multi-criteria decision algorithm can help overcome the limitations that arise when only one of the mentioned object representations is used or when both object representations are combined, as in the above-mentioned work [25]. In a method proposed in [18] a multi-criteria analysis is leveraged to enhance 6D object pose tracking in RGB videos on the basis of object keypoints and object shape.

Motivated by the need for robust methods capable of dealing with uncertain and ambiguous visual data, in this work, we propose a pipeline for 6D object pose tracking on RGB images, wherein a fuzzy TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) is employed. The proposed method calculates decision variables on features that are frequently utilized in relevant work including segmented masks, fiducial keypoints, and the distance transform (DT). The decision variables are determined taking into account the object's non-occluded part and the object's occluded part. We propose a two-stack hourglass network to jointly estimate nine objects keypoints and predict both the object's non-occluded part and the object's occluded part. We discuss the proposed loss function and training the multi-task hourglass. To verify our designs, we conduct thorough experiments on the challenging YCB-Video benchmark dataset. We demonstrate experimentally that the proposed algorithm achieves competitive results.

The rest of the paper is organized as follows. In the next Section we present the proposed method. At the beginning we outline the background. Afterwards, we present the proposed multi-task hourglass neural network. Finally, we present fuzzy TOPSIS aided pose tracking. In Section III, after outlining metrics and dataset, we present experimental results. The sections end with an ablation study. In Section IV we present conclusions.

## II. PROPOSED METHOD

### A. BACKGROUND

Multicriteria decision making (MCDM) permits selecting a finite number of alternatives characterized by multiple conflicting criteria. TOPSIS [13] allows evaluation of the performance of alternatives by similarity to the ideal solution. According to this algorithm, the geometric distance of the lowest-ranked alternative is the nearest to the worst solution while the highest-ranked alternative is the closest to the ideal solution. Design an MCDM system comprises alternatives and criteria, which form a decision matrix. On the basis of information in the form of the decision matrix and criteria weights, TOPSIS ranks all the alternative candidates. As the data of the decision matrix usually are delivered by different sources, so it is necessary to normalize it, which permits comparisons of various criteria. The normalization converts the elements of the decision matrix into a non-dimensional form. The goal of normalization techniques is to scale the elements of the decision matrix to be approximately of the same magnitude.

In many real life decision making problems, data from observations are often very uncertain or imprecise, including visual ones. Because MCDM systems may not be as effective as they could be in dealing with the imprecise or unclear nature of data, they are very often designed using fuzzy set theory and fuzzy logic. A fuzzy number can be seen as an extension of an interval with a varied grade of membership. This means that each value in the interval has associated a real number that indicates its compatibility with the vague statement associated with a fuzzy number. Fuzzy numbers have their own rules of operation. In practice, the triangular shape of the membership function is often

employed to represent fuzzy numbers. The distance from the positive-ideal solutions and the negative-ideal solutions can be computed using several distance metrics. In most extensions of fuzzy TOPSIS, decisions are determined using Chen's vertex method [5].

### B. MULTI-TASK HOURGLASS NEURAL NETWORK

The goal of object pose estimation is to predict the 3D position and 3D rotation of the object of interest in camera-centered coordinates. Object pose estimation is usually done on single images. Algorithms for object pose tracking take advantages of the temporal consistency among video frames. They leverage information from the previous frame to enhance recovering object pose in image sequences [10], [26]. Existing object pose tracking algorithms typically take only the current frame and the previous frame as input to predict the pose in the current frame [10]. Estimating the 6D object pose on RGB images is a very challenging task. The main difficulty is recovering mapping the object from RGB images to 3D space. The discussed task still poses challenges due to textureless appearances, occlusions, and object symmetries.

The proposed algorithms operate on RGB images that are acquired by a calibrated camera. We assume that the object to be tracked is rigid and its 3D model is available. A mesh model is defined by a set of 3D vertices, edges, and triangular faces. As our algorithms employ information from the former frame they track 6D object pose on image sequences. Input RGB images are fed to our hourglass-based neural network, which jointly detects nine object keypoints, predicts the object's non-occluded part, and also predicts object's occluded part, see Fig. 1. The network detects nine key points of the object regardless of its pose, including poses in which some keypoints are on the invisible side of the object or in the case of partial occlusions of the object. The values of the heatmaps representing locations of the object keypoints are fed to the TOPSIS algorithm. Aside from the information on the estimated object keypoints our TOPSIS aided algorithms employ also information about object segmentation, distance transform, and keypoints detected in the previous frame. This way we utilize temporal consistency among video frames. Our TOPSIS aided algorithms, which were designed to cope with imperfect and uncertain attributes remove the worst keypoint out of nine object keypoints through ranking various alternatives. Finally, the object pose is estimated by the PnP algorithm in combination with RANSAC-based scheme. Information about the object location in the image is utilized to crop the object in the next input frame, see Fig. 1.

In this work, our TOPSIS-based algorithms operate on four decision variables. Fig. 2 illustrates the main steps in determining decision variables. The decision variables that are determined for each object keypoint are as follows:

- Value of projected 3D keypoint on the corresponding heat map, which is determined by the hourglass neural network. The higher the value is, the better.

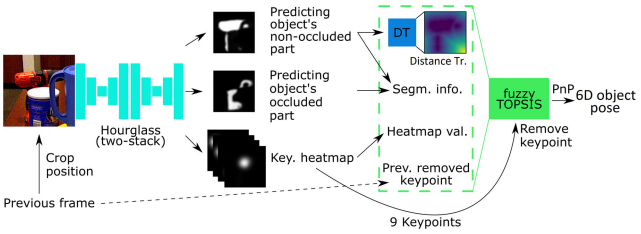


**FIGURE 1. Overview of our TOPSIS aided algorithm for 6D object tracking in sequences of RGB images.**

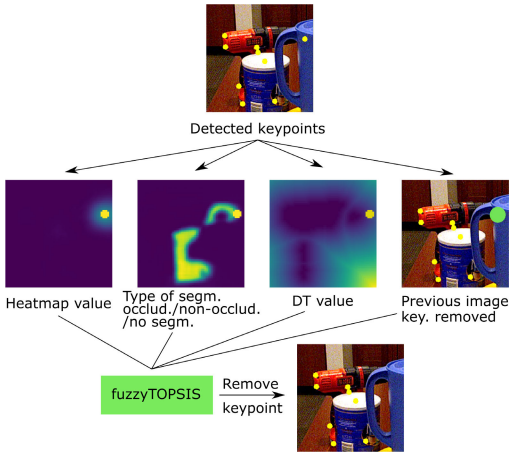


**FIGURE 2. Decision variables used in TOPSIS.**

- Value depending on type of the object segmentation on which the keypoint is located. If the keypoint is located on the image background the variable assumes a big value, if it is located on the occluded segment of the object the variable assumes a middle value, whereas in case it is located on the visible segment of the object the variable assumes a small value. The smaller the value is, the better.
- Value depending on the distance transform at which the keypoint is located to the closest visible pixel. The smaller the value of the distance transform at the location of the keypoint, the better.
- Value depending on whether the keypoint was omitted in the previous frame or was used by the PnP to determine the pose. If the keypoint was not omitted the variable assumes a higher value, otherwise the variable assumes a smaller value. The higher the value is, the better.

We utilize the hourglass (HG) network [20] as the backbone of our multi-task neural network. An hourglass module is a type of encoder-decoder network. It first downsamples the input features, then upsamples them to the original size, where skip layers are used to hold details in the upsampled features. Thanks to this, the HG module is able to capture both global and local features. By stacking multiple HG modules the latter HG module can learn from the intermediate predictions of the previous module, i.e. the later module reprocesses the intermediate predictions to capture higher-level of information. The architecture of

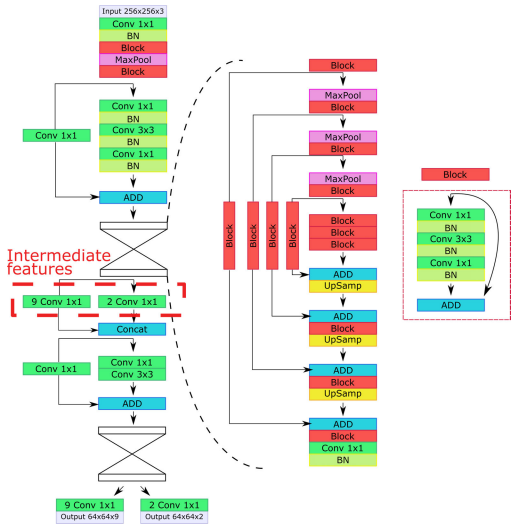


**FIGURE 3.** Two-stack hourglass neural network for jointly object keypoints detection, predicting the visible (not occluded) object part and predicting occluded part of the object.

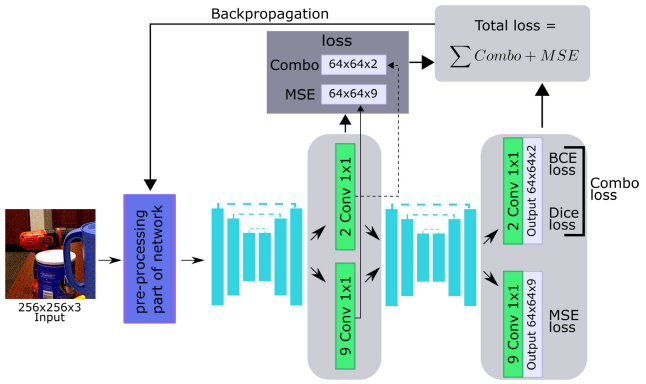


**FIGURE 4.** Details of training of multi-task hourglass for joint object's keypoint estimation and object's occluded/non-occluded part segmentation.

our hourglass-based multi-task neural network is shown in Fig. 3. It was designed to jointly detect nine object keypoints, predict the object's non-occluded part, and to predict the object's occluded part. The neural network operates on RGB images of size 256 × 256 and delivers the object's non-occluded part and the object's occluded part on two separate output map channels. On nine output maps the neural network generates blobs, whose centers represent 2D positions of the object keypoints. Each object keypoint is generated on a specific channel to provide 2D-3D correspondences. When training the network, the keypoints were represented by two-dimensional Gaussian functions with $\sigma$ equal to five pixels and centered on the desired positions on the object.

The proposed neural network consists of two hourglass blocks, see Fig. 3. Each basic block, see also red block on Fig. 3, consists of two branches: one with $1 \times 1$, $3 \times 3$, and $1 \times 1$ convolutional blocks followed by batch normalization, and the second one with a direct connection to calculate the residual. After the first part of basic blocks and max pooling, a residual block that is similar to the basic block except that instead of a direct connection a $1 \times 1$ convolution is utilized in order to calculate the residual. The first hourglass block is followed by a residual block with a $1 \times 1$ convolution followed by a $3 \times 3$ convolution in the first branch and a $1 \times 1$ convolution in the second one.

Typically, neural networks are designed to perform a single task. The core idea behind multi-task learning [4] is that owing to sharing encoded information between tasks, the learned models for different tasks will be similar to each other, and such multi-task training will be improved over independent training of individual tasks, particularly in case of a limited amount of training data. Our hourglass model executes multiple tasks, i.e. estimates keypoints, and detects both occluded and non-occluded parts of the object. Heatmap regression and coordinate regression are two commonly used approaches in neural network-based landmark localization. Heatmap-based (also known as confidence map-based) methods aim at predicting heatmaps such that points with maximum values correspond to keypoints in the input image. In our approach the real keypoints are represented by Gaussian heatmaps, and the heatmap regression model was optimized with the pixel-wise MSE loss. The segmentation model was optimized using a Combo loss that is a sum of binary cross-entropy (BCE) loss and Dice loss. Fig. 4 details the process of training the proposed multi-task hourglass for joint object's keypoints estimation, object's non-occluded part segmentation, and object's occluded part segmentation. During the calculation of the loss, both feature maps determined by two-stack hourglass blocks were utilized.

### C. FUZZY TOPSIS AIDED POSE TRACKING

In this work, two versions of TOPSIS aided algorithms have been implemented and evaluated. In the first version, the TOSIS operates on crisp values while in the second version, a fuzzy TOPSIS has been used. This section introduces the basic definitions and the procedure of the fuzzy TOPSIS method. Then, the proposed method is illustrated with numerical examples. The presented notation was based on publication [14] and only a single decision-maker is utilized in obtaining the appropriate decision. The decision-maker has to choose one of $m$ possible alternatives $a_i$ described by $n$ criteria $C_j$ measured by triangle fuzzy numbers and linguistic values.

Let:

$$\tilde{w} = (\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_n) \tag{1}$$

be the vector of criteria weights, in which $w_j$ is the weight of criterion $C_j$. This vector allows to emphasize more important criteria and to reduce the influence of less important ones. The set of criteria is divided into two subsets: $BC$ criteria (benefit criteria) whose greater value is better and $CC$ criteria (cost criteria) whose lower value is better. A fuzzy multi-criteria

problem can be concisely expressed in matrix form as:

$$X = \begin{array}{c} \\ a_1 \\ a_2 \\ \\ a_m \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & & C_n \end{array} \\ \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & x_{ij} & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix} \end{array} \quad (2)$$

The matrix consists of $m$ possible alternatives $a_i$ described by $n$ criteria $C_j$. In fuzzy set theory, conversion scales are applied to transform linguistic terms into fuzzy numbers. The construction of a conversion scale is discussed for example by [1]. A linguistic variable is a variable whose value is given in linguistic terms [40]. The acquired linguistic values are then transformed according to the appropriate linguistic terms table into triangle fuzzy numbers $\tilde{e}_{ij} = (e^a_{ij}, e^b_{ij}, e^c_{ij})$ to get the processed decision matrix $E$.

$$E = \begin{array}{c} \\ a_1 \\ a_2 \\ \\ a_m \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & & C_n \end{array} \\ \begin{pmatrix} \tilde{e}_{11} & \tilde{e}_{12} & \dots & \tilde{e}_{1n} \\ \tilde{e}_{21} & \tilde{e}_{22} & \dots & \tilde{e}_{2n} \\ \dots & \dots & \tilde{e}_{ij} & \dots \\ \tilde{e}_{m1} & \tilde{e}_{m2} & \dots & \tilde{e}_{mn} \end{pmatrix} \end{array} \quad (3)$$

Then, a normalized decision matrix $Z$ should be constructed using matrix $E$. The ranges of normalized triangular fuzzy numbers which belong to [0,1] are preserved by the linear normalization method. For the benefit criteria, normalization is expressed as follows:

$$\tilde{z}_{ij} = \left(\frac{e^a_{ij}}{e^+_j}, \frac{e^b_{ij}}{e^+_j}, \frac{e^c_{ij}}{e^+_j}\right), \text{where } \tilde{e}_{ij} \in BC, e^+_j = \max_i(e^c_{ij}) \quad (4)$$

While for the cost criteria, the normalization is performed as follows:

$$\tilde{z}_{ij} = \left(\frac{e^-_j}{e^a_{ij}}, \frac{e^-_j}{e^b_{ij}}, \frac{e^-_j}{e^c_{ij}}\right), \text{where } \tilde{e}_{ij} \in CC, e^-_j = \min_i(e^a_{ij}) \quad (5)$$

where $BC$ and $CC$ denote beneficial and cost criteria, respectively. Thus, the normalized decision matrix $Z$ is created:

$$Z = \begin{array}{c} \\ a_1 \\ a_2 \\ \\ a_m \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & & C_n \end{array} \\ \begin{pmatrix} \tilde{z}_{11} & \tilde{z}_{12} & \dots & \tilde{z}_{1n} \\ \tilde{z}_{21} & \tilde{z}_{22} & \dots & \tilde{z}_{2n} \\ \dots & \dots & \tilde{z}_{ij} & \dots \\ \tilde{z}_{m1} & \tilde{z}_{m2} & \dots & \tilde{z}_{mn} \end{pmatrix} \end{array} \quad (6)$$

The weighted normalized fuzzy decision matrix $V$, consisting of $\tilde{v}_{ij}$, is computed by multiplying the normalized matrix $Z$ with the criteria weight, see (1), as follows:

$$\tilde{v}_{ij} = \tilde{z}_{ij} \times \tilde{w}_j \quad (7)$$

where $\times$ is a fuzzy product [15], $\tilde{w}_j$ is the fuzzy importance weight for criterion $j$ and in a scenario involving more than one decision-maker in categorizing the degree of importance of criteria, it is referred to as combined group criteria weight [9]. Due to the use of a single decision-maker in the decision-making process, the values of criteria weights are not aggregated from many decision-makers, as it is conventionally done, but are chosen by one single decision-maker from the linguistic terms table.

The positive ideal solution $A^+$ and the negative ideal solution $A^-$ can be attained from the weighted normalized decision matrix $V$. The selection of $A^+$ and $A^-$ is performed as follows:

$$A^+ = (\tilde{v}^+_1, \tilde{v}^+_2, \dots, \tilde{v}^+_n), \text{where } \tilde{v}^+_j = \max_i(\tilde{v}_{ij}) \quad (8)$$

$$A^- = (\tilde{v}^-_1, \tilde{v}^-_2, \dots, \tilde{v}^-_n), \text{where } \tilde{v}^-_j = \min_i(\tilde{v}_{ij}) \quad (9)$$

As fuzzy triangle numbers are used in Eq. (8-9), the values of the three fuzzy numbers $\tilde{v}^+_j = (v^{a+}_j, v^{b+}_j, v^{c+}_j)$ are determined independently of each other, and $\max_i(\tilde{v}_{ij})$ is calculated as follows:

$$\max_i(\tilde{v}_{ij}) = (\max_i v^a_{ij}, \max_i v^b_{ij}, \max_i v^c_{ij}) \quad (10)$$

where $\max_i v^a_{ij}$ denotes maximum value for all $v^a_i$ numbers for criterion $j$. Similarly for $\tilde{v}^-_j = (v^{a-}_{ij}, v^{b-}_{ij}, v^{c-}_{ij})$, the formula $\min_i(\tilde{v}_{ij})$ is calculated as follows:

$$\min_i(\tilde{v}_{ij}) = (\min_i v^a_{ij}, \min_i v^b_{ij}, \min_i v^c_{ij}) \quad (11)$$

Applying the vertex method, the distances between each alternative $a_m$, acquired by (7), and $A^+$ can be determined. Since $\tilde{v}_{ij} = (v^a_{ij}, v^b_{ij}, v^c_{ij})$ and $\tilde{v}^+_j$ from (8) can be described as $\tilde{v}^+_j = (v^{a+}_j, v^{b+}_j, v^{c+}_j)$, the distance can be expressed as follows:

$$d(\tilde{v}_{ij}, \tilde{v}^+_j) = \times \sqrt{\frac{1}{3}[(v^a_{ij} - v^{a+}_j)^2 + (v^b_{ij} - v^{b+}_j)^2 + (v^c_{ij} - v^{c+}_j)^2]} \quad (12)$$

Similarly, the distances between each alternative and $A^-$ can be expressed as follows:

$$d(\tilde{v}_{ij}, \tilde{v}^-_j) = \times \sqrt{\frac{1}{3}[(v^a_{ij} - v^{a-}_j)^2 + (v^b_{ij} - v^{b-}_j)^2 + (v^c_{ij} - v^{c-}_j)^2]} \quad (13)$$

Therefore, the distance of each alternative to $A^+$ and $A^-$ can be determined in the following manner:

$$d^+(a_i) = \sum_{j=1}^{n} d(\tilde{v}_{ij}, \tilde{v}^+_j), \ i = 1, 2, .., m; j = 1, 2.., n \quad (14)$$

$$d^-(a_i) = \sum_{j=1}^{n} d(\tilde{v}_{ij}, \tilde{v}^-_j), \ i = 1, 2, .., m; j = 1, 2.., n \quad (15)$$

The closeness degree of each alternative $a_i$ is defined as:

$$D_i = \frac{d^-(a_i)}{d^+(a_i) + d^-(a_i)} \quad (16)$$

The closeness determines the ranking order and indicates the best and worst of all alternatives.

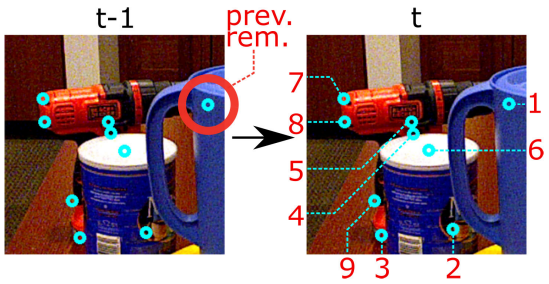


**FIGURE 5.** Deciding the worst keypoint shown in the toy example.

**TABLE 1.** The values of decision variables for each keypoint acquired from hourglass and previous frame. Each keypoint is one of the alternatives $A$.

| | $C_1$ : heatmap | $C_2$ : segm. | $C_3$ : DT | $C_4$ : prev. rem. |
|---|---|---|---|---|
| $a_1$ | 239 | occluded | 6 | yes |
| $a_2$ | 207 | occluded | 2 | no |
| $a_3$ | 251 | occluded | 1 | no |
| $a_4$ | 226 | occluded | 1 | no |
| $a_5$ | 242 | non-occl. | 1 | no |
| $a_6$ | 248 | occluded | 5 | no |
| $a_7$ | 251 | non-occl. | 1 | no |
| $a_8$ | 253 | non-occl. | 1 | no |
| $a_9$ | 239 | non-occl. | 1 | no |

Below we present a numerical example using toy data prepared using the YCB-Video dataset. Fig. 5 presents a toy example in which a fuzzy TOPSIS algorithm is prepared and executed in order to find the worst keypoint to remove for the pose tracking in frame $t$. To determine the worst keypoint, first, an fuzzy decision matrix with fuzzy weights needs to be prepared, see (2). Weights, see (1), indicate which criteria should have a more significant impact on the final decision. The matrix consists of $m$ possible alternatives $a_i$ described by $n$ criteria $C_j$. Each keypoint is represented by one alternative, therefore, $m = 9$, whereas $n = 4$, because four values were acquired from the neural network and previous frames.

The decision variables, which are shown in Tab. 1, are the values of the heatmap for projected 3D keypoints, the type of segmentation on which the keypoint is located, the value depending on the distance transform, and a value depending on whether the keypoint was omitted in the previous frame. The $C_1$ and $C_4$ criteria belong to the $BC$ subset whereas the $C_2$ and $C_3$ criteria belong to the $CC$ subset. It means that the first two are expected to have a high value and the second two a low value. The $C_2$ and $C_4$ criteria are in a linguistic form.

To calculate fuzzy weights, see (1), due to only one decision-maker, linguistic variables are chosen from the linguistic terms table designed by us, see Tab. 2, and then changed into corresponding fuzzy triangular numbers.

The corresponding fuzzy triangular numbers for segmentation linguistic variable and previously removed keypoints linguistic variable are shown in Tab. 3 and Tab. 4, respectively. The decision variables shown in Tab. 1 are converted into fuzzy numbers to construct the fuzzy decision matrix, see Tab. 5.

**TABLE 2.** Linguistic variables for the relative importance weights of criteria.

| Linguistic variable | Fuzzy triangular number |
|---|---|
| Of little importance | (0.00, 0.00, 0.20) |
| Moderately important | (0.05, 0.25, 0.45) |
| Important | (0.30, 0.50, 0.70) |
| Very important | (0.55, 0.75, 0.95) |
| Absolutely important | (0.80, 1.00, 1.00) |

**TABLE 3.** Linguistic variables for the segmentation criteria.

| Linguistic variable | Fuzzy triangular number |
|---|---|
| Non-occluded | (1, 1, 3) |
| Occluded | (1, 2, 3) |
| Background | (1, 3, 3) |

**TABLE 4.** Linguistic variables for the previously removed keypoint criteria.

| Linguistic variable | Fuzzy triangular number |
|---|---|
| Yes | (1, 1, 2) |
| No | (1, 2, 2) |

The fuzzy triangular numbers for $C_1$ and $C_3$ are generated using information about the maximal and minimal values from the hourglass, which are different for each $a_i$ due to separate channels for heatmaps. For each frame, the values vary slightly. The acquired fuzzy numbers for criteria weights and criteria for each alternative are shown in Tab. 5.

The (normalized) fuzzy decision matrix is constructed using (4) and (5), c.f. Tab. 6.

The normalization method mentioned above is to preserve the property that the ranges of normalized triangular fuzzy numbers belong to [0,1]. Considering the different importance of each criterion, the weighted normalized fuzzy decision matrix is constructed using Eq. (7), see Tab. 7.

Using positive ideal solution (8) and negative ideal solution (9), and Eq. (16), the calculation of the relative closeness is shown in Tab. 8. According to the relative coefficient, we can determine the ranking order of all alternatives. We experimented also with the following distances: Euclidean, weighted Euclidean, Hamming [2], weighted Hamming, and L-R distance [32]. However, the improvement in performance was not statistically significant.

## III. EXPERIMENTAL RESULTS

First, we outline metrics and afterwards we describe the YCB-Video dataset. Next, we present the performance of our algorithms along with the performance of state-of-the-art algorithms. We conclude the Section with an ablation study to determine which components of the proposed algorithm contribute most to performance improvements.

### A. METRICS

The quality of 6D object pose tracking was determined using ADD (Average Distance of Model Points) score [11] as metric for non-symmetric objects and Average Closest Point Distance (ADD-S) for symmetric ones. For

**TABLE 5.** The fuzzy decision matrix and fuzzy weights.

| | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
|---|---|---|---|---|
| $w$ | (0.05, 0.25, 0.45) | (0.55, 0.75, 0.95) | (0.05, 0.25, 0.45) | (0.3, 0.5, 0.7) |
| $a_1$ | (178, 239, 239) | (1, 2, 3) | (1, 6, 21) | (1, 1, 2) |
| $a_2$ | (160, 207, 207) | (1, 2, 3) | (1, 2, 21) | (1, 2, 2) |
| $a_3$ | (200, 251, 251) | (1, 2, 3) | (1, 1, 21) | (1, 2, 2) |
| $a_4$ | (169, 226, 232) | (1, 2, 3) | (1, 1, 21) | (1, 2, 2) |
| $a_5$ | (191, 242, 243) | (1, 1, 3) | (1, 1, 21) | (1, 2, 2) |
| $a_6$ | (182, 248, 251) | (1, 2, 3) | (1, 5, 21) | (1, 2, 2) |
| $a_7$ | (189, 251, 253) | (1, 1, 3) | (1, 1, 21) | (1, 2, 2) |
| $a_8$ | (176, 253, 253) | (1, 1, 3) | (1, 1, 21) | (1, 2, 2) |
| $a_9$ | (177, 239, 241) | (1, 1, 3) | (1, 1, 21) | (1, 2, 2) |

**TABLE 6.** The normalized fuzzy decision matrix.

| | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
|---|---|---|---|---|
| $a_1$ | (0.7, 0.9, 0.9) | (0.3, 0.5, 1.0) | (0.04, 0.16, 1.0) | (0.5, 0.5, 1.0) |
| $a_2$ | (0.6, 0.8, 0.8) | (0.3, 0.5, 1.0) | (0.04, 0.5, 1.0) | (0.5, 1.0, 1.0) |
| $a_3$ | (0.7, 0.9, 0.9) | (0.3, 0.5, 1.0) | (0.04, 1.0, 1.0) | (0.5, 1.0, 1.0) |
| $a_4$ | (0.6, 0.8, 0.9) | (0.3, 0.5, 1.0) | (0.04, 1.0, 1.0) | (0.5, 1.0, 1.0) |
| $a_5$ | (0.7, 0.9, 0.9) | (0.3, 1.0, 1.0) | (0.04, 1.0, 1.0) | (0.5, 1.0, 1.0) |
| $a_6$ | (0.7, 0.9, 0.9) | (0.3, 0.5, 1.0) | (0.04, 0.18, 1.0) | (0.5, 1.0, 1.0) |
| $a_7$ | (0.7, 0.9, 1.0) | (0.3, 1.0, 1.0) | (0.04, 1.0, 1.0) | (0.5, 1.0, 1.0) |
| $a_8$ | (0.6, 1.0, 1.0) | (0.3, 1.0, 1.0) | (0.04, 1.0, 1.0) | (0.5, 1.0, 1.0) |
| $a_9$ | (0.7, 0.9, 0.9) | (0.3, 1.0, 1.0) | (0.04, 1.0, 1.0) | (0.5, 1.0, 1.0) |

ADD score, the object pose is considered correct if the average vertex-to-vertex distance in 3D space, i.e. distance between the 3D vertices transformed on the basis of the estimated pose and pose calculated on the basis of the ground-truth is below $0.1d$, where $d$ stands for the object diameter. The ADD can be expressed in the following manner:

$$\text{ADD} = \text{avg}_{x \in M} ||(R\,x + t) - (\hat{R}\,x + \hat{t})||_2 \qquad (17)$$

where $M$ stands for a set of 3D object model points, $x$ stands for vertices randomly selected from the 3D model of the considered object, $t$ and $R$ stand for the translation and rotation of the ground truth transformation, respectively, whereas $\hat{t}$ and $\hat{R}$ are the predicted translation and rotation, respectively. For the YCB-Video dataset, the ADD-S [11], [34] was employed to measure the pose error. It is calculated as follows:

$$\text{ADD-S} = \text{avg}_{x_1 \in M} \min_{x_2 \in M} ||(R\,x_1 + t) - (\hat{R}\,x_2 + \hat{t})||_2 \qquad (18)$$

where $x_1$ and $x_2$ are selected from the 3D object's model. For evaluations on the YCB-Video dataset we further calculated the AUC (area under curve) of ADD/ADD-S by varying the distance threshold from 0m to 0.1m as in work that introduced the PoseCNN [34].

### B. DATASET

The proposed algorithms have been evaluated on YCB-Video benchmark that is large-scale dataset for 6D object pose estimation [34]. The discussed YCB-Video dataset comprises objects of various shapes and texture levels under different occlusion and illumination conditions. Image sequences may contain multiple target objects. This benchmark dataset provides accurate 6D poses of 21 objects, which were observed in 92 videos with 133 827 frames. We followed the approach presented in [34] to calculate the AUC ADD scores under the accuracy-threshold curve obtained by varying the distance threshold with a maximum threshold equal to 10 centimeters. According to the recommendations mentioned above 80 video sequences are used for training the models, and 2 949 key frames extracted from the remaining twelve videos are employed for evaluations of algorithms.

**TABLE 7.** The weighted normalized fuzzy decision matrix.

| | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
|---|---|---|---|---|
| $a_1$ | (0.03, 0.23, 0.42) | (0.18, 0.37, 0.95) | (0.0, 0.04, 0.45) | (0.15, 0.25, 0.7) |
| $a_2$ | (0.03, 0.20, 0.36) | (0.18, 0.37, 0.95) | (0.0, 0.12, 0.45) | (0.15, 0.5, 0.7) |
| $a_3$ | (0.04, 0.24, 0.44) | (0.18, 0.37, 0.95) | (0.0, 0.24, 0.45) | (0.15, 0.5, 0.7) |
| $a_4$ | (0.03, 0.22, 0.41) | (0.18, 0.37, 0.95) | (0.0, 0.24, 0.45) | (0.15, 0.5, 0.7) |
| $a_5$ | (0.03, 0.23, 0.43) | (0.18, 0.75, 0.95) | (0.0, 0.24, 0.45) | (0.15, 0.5, 0.7) |
| $a_6$ | (0.03, 0.24, 0.44) | (0.18, 0.37, 0.95) | (0.0, 0.04, 0.45) | (0.15, 0.5, 0.7) |
| $a_7$ | (0.03, 0.24, 0.45) | (0.18, 0.75, 0.95) | (0.0, 0.24, 0.45) | (0.15, 0.5, 0.7) |
| $a_8$ | (0.03, 0.25, 0.45) | (0.18, 0.75, 0.95) | (0.0, 0.24, 0.45) | (0.15, 0.5, 0.7) |
| $a_9$ | (0.03, 0.23, 0.42) | (0.18, 0.75, 0.95) | (0.0, 0.24, 0.45) | (0.15, 0.5, 0.7) |

**TABLE 8.** The relative closeness of the keypoints. In this case, the worst selection is candidate $a_1$.

| | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ | $a_7$ | $a_8$ | $a_9$ |
|---|---|---|---|---|---|---|---|---|---|
| $D$ | 0.35 | 0.36 | 0.38 | 0.37 | 0.40 | 0.37 | 0.41 | 0.41 | 0.40 |

### C. EXPERIMENTAL RESULTS

The hourglass neural network has been trained in 1000 epochs with a batch size set to 16, using RMSprop (Root Mean Square propagation) with a learning rate set to 1e-4. For each object in the YCB-Video dataset a single neural network was trained. The training subset was expanded with 80k synthetically rendered images released in the dataset. Due to lack of information about occluded parts of the objects in the YCB-Video dataset it has been extended about masks delivering such information using ground-truth, existing visible object segmentations, and the projected 3D object models onto the images with masks.

In all experimental evaluations, we followed the recommended dataset divisions. The performance of our algorithms along with the performance of state-of-the-art algorithms is provided in Tab. 9. The AUC ADD scores achieved by our algorithms are compared with results attained by algorithms operating on single frame as well as algorithms for 6D object pose tracking. The bold result in each row in the table indicates the highest AUC ADD value for the given object, whereas the second best result is underlined. It can be observed that the TOPSIS aided algorithm permits achieving far better results in comparison to a base algorithm that determines the object pose on the basis of sparse object keypoints and the PnP in combination with RANSAC-based scheme. As we can see, the results achieved by our fuzzy TOPSIS-based algorithm are far better than the results attained by our TOPSIS aided algorithm.

Afterwards, we implemented a simple voting-based decision algorithm. The worst keypoint is removed using the recently proposed Borda rule on top-truncated preferences [28]. In the proposed approach the rules are truncated to

two worst preferences. It turned out that using a voting algorithm that is based on ordinary Borda count often leads to a selection of a keypoint that was never at the first place, but obtained the best average ratings. It is clear that keypoints with the best average ratings not necessarily may lead to optimal selections of keypoints, but rather to smoothed trajectories of object poses. In addition to fuzzy TOPSIS, recently proposed fuzzy MABAC [30] and fuzzy MAIRCA [8] were used in our voting-based decision committee (called VDC-fuzzyTOPSIS). By comparing results in the last two columns of Tab. 9, we can see that owing to using diverse algorithms for multi-criteria decision making with Borda count–based voting scheme it is possible to improve the AUC ADD scores. The value 128 indicates that the results were achieved by an hourglass neural network delivering output maps of size $128 \times 128$. The basis version of the hourglass network delivers output maps of size $64 \times 64$, c.f. Fig. 3. Given the results in discussed table it is evident that TOPSIS algorithms operating on features that are frequently utilized in algorithms for object tracking including segmented masks, sparse keypoints, and distance transform permit achieving tangible performance gains. It is worth noting that for several objects from YCB-Video dataset our VDC-fuzzyTOPSIS algorithm achieved the best and second best results. Fig. 6 presents qualitative results on the YCB-Video dataset.

Tab. 10 compares results achieved by our algorithm with results of recent algorithms estimating the object pose on the single frame [21], [29], [31], [34], [37] with results achieved by tracking-based methods [6], [16]. As we can observe, the AUC ADD scores achieved by DeepIm and our algorithm are better than the scores that were achieved by PoseRBPF and all algorithms that estimate the pose without taking into account information from the previous frame, i.e. on the basis of a single frame. DeepIM operating on single frames achieves far worse AUC ADD scores compared to the results obtained by our algorithm. One of the main reasons of worse results achieved by DeepIm is that it employs the FlowNetSimple [7] to estimate optical flow between two successive images, which is used to predict a relative SE(3) transformation between the observed and rendered object maps.

Dealing with occlusions is an important point in 6D object pose estimation. Since most objects in the visual world are partially obscured, this problem is encountered when estimating the object pose in most tasks. Tab. 11 presents MSE errors for object keypoints and Dice scores for object segmentation, which were attained by the proposed hourglass-based neural network, which jointly detects nine object keypoints, predicts the object's non-occluded part, and also predicts the object's occluded part. In order to demonstrate the potential of neural network in the detection of occluded parts of the object the Dice scores are presented separately for predicting object's non-occluded part and predicting object's occluded part. As we can observe, promising results have been achieved for selected objects from the YCB-Video dataset. To the best of our knowledge, this occlusion-aware is the first pipeline in object pose estimation that jointly detects objects keypoints and recovers non-occluded/occluded object parts and then effectively leverages them to improve the accuracy of pose estimation. Our approach differs from [22] since instead of a GAN network an hourglass neural network is used to jointly detect object keypoints, predict the object's non-occluded part, and to predict the object's occluded part, and then utilize such information in multi-criteria decision making, where the uncertainty of observations and predictions is taken into account.

### D. ABLATION STUDY

In this Section, we analyze the efficacy of TOPSIS-based components and prove the superiority of our algorithm over the baseline one. We inspect the impact of adding TOPSIS, and fuzzy TOPSIS. We analyze the effect that our extensions have on the ADD scores.

The experiments have been conducted on the power drill object from the 56th test video. Fig. 8 presents a plot of the occlusion degree of the power drill vs. frame number. As it can be observed, at the beginning of the sequence, the object of interest is the most occluded and with each frame the degree of the occlusion gradually decreases. The change in the pose of the object between successive frames is small, particularly, there are no jerks, and the total changes in the object pose are not too large, i.e. up to 40 degrees over the entire sequence, see also Fig. 7. As it can be observed, frames from the beginning of the sequence are better suited for achieving high ADD scores, even though the object is partially occluded, because the wide side of the object is observed by the camera. On the other hand, the frames at the end of the sequence are less suitable for high ADD scores because the camera observes the narrower side of the object. The second plot presents ADD score errors vs. frame number that have been achieved by the PnP algorithm operating on nine object keypoints. The ADD score on the whole video is equal to 0.9029. As it can be observed, the biggest errors are in the beginning part of the sequence. The next plot presents ADD score errors over time that have been attained by the PnP algorithm operating on eight object keypoints. In the discussed experiment the hourglass detected nine object keypoints and then the worst object keypoint has been removed by TOPSIS. This means that eight object keypoints selected in such a way were fed to the PnP algorithm with the RANSAC scheme [3] for dealing with outliers. The ADD score achieved by such TOPSIS aided algorithm was equal to 0.9584. As it can be observed, the use of TOPSIS permitted to achieve a far better ADD score. The last plot presents ADD score errors over time that have been attained by the PnP algorithm operating on eight object keypoints, which were selected by the fuzzy PnP. The ADD score achieved by such fuzzy TOPSIS aided algorithm was equal to 0.9837. It can therefore be concluded that both TOPSIS algorithms permit achieving better ADD scores, but the fuzzy TOPSIS yields

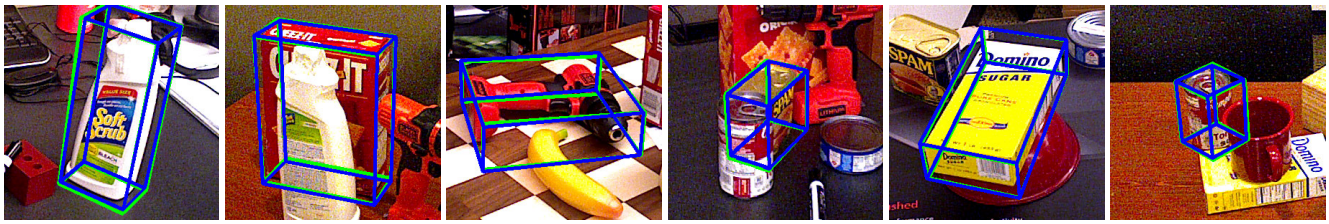

**FIGURE 6.** Qualitative results on the YCB-Video dataset. 021_bleach_cleanser, 003_cracker_box, 035_power_drill, 010_potted_meat_can, 004_sugar_box, and 005_tomato_soup_can. The green bounding boxes show the ground truth poses, while the blue ones correspond to the estimated poses. For a better visualization, we cropped regions of interest.

**TABLE 9.** Comparison of AUC ADD scores [%] (max. th. 10 cm) of the proposed method with some state-of-the-art methods on the YCB-Video dataset. AUC ADD is calculated for non-symmetric objects, whereas AUC ADD-S is determined for symmetric objects, '-' denotes unavailable results, and '*' stands for symmetric objects.

| Object | PoseRBPF | DeepIM | [18] | PnP | TOPSIS | fuzzy TOPSIS | Our VDC-fuzzyTOPSIS | Our VDC-fuzzyTOPSIS 128 |
|---|---|---|---|---|---|---|---|---|
| 002_master_chef_can | 63.3 | 89.0 | **90.2** | 87.7 | 88.7 | 90.1 | **90.2** | 88.8 |
| 003_cracker_box | 77.8 | **88.5** | 85.4 | 83.3 | 86.4 | 86.7 | 87.4 | 87.1 |
| 004_sugar_box | 79.6 | **94.3** | 87.5 | 82.7 | 85.7 | 86.2 | 88.4 | 89.3 |
| 005_tomato_soup_can | 73.0 | **89.1** | 82.3 | 81.5 | 81.7 | 82.5 | 83.5 | 83.4 |
| 006_mustard_bottle | 84.7 | **92.0** | 90.0 | 81.9 | 83.7 | 86.3 | 87.0 | 89.2 |
| 007_tuna_fish_can | 64.2 | **92.0** | 87.9 | 80.3 | 81.7 | 82.6 | 84.4 | 90.4 |
| 008_pudding_box | 64.5 | 80.1 | 83.3 | 80.0 | 82.4 | **88.2** | 84.4 | 87.1 |
| 009_gelatin_box | 83.0 | 92.0 | **94.4** | 83.4 | 87.9 | 89.2 | 90.4 | 90.2 |
| 010_potted_meat_can | 51.8 | 78.0 | **82.3** | 73.1 | 74.4 | 74.8 | 81.3 | 81.0 |
| 011_banana | 18.4 | **81.0** | 73.6 | 63.6 | 67.4 | 67.0 | 67.5 | 78.1 |
| 019_pitcher_base | 63.7 | 90.4 | **90.7** | 84.2 | 85.1 | 86.7 | 89.4 | 89.4 |
| 021_bleach_cleanser | 60.5 | 81.7 | 80.6 | 77.3 | 78.8 | 79.9 | 81.7 | **82.6** |
| 024_bowl* | 85.6 | 90.6 | 87.9 | 84.1 | 89.3 | 90.5 | 91.0 | **91.1** |
| 025_mug | 77.9 | 83.2 | 85.8 | 80.5 | 83.5 | 83.7 | 84.7 | **86.6** |
| 035_power_drill | 71.8 | **85.4** | 79.7 | 78.6 | 81.3 | 81.9 | 83.3 | 84.7 |
| 036_wood_block* | 31.4 | 75.4 | **82.6** | 77.1 | 77.6 | 79.3 | 81.7 | 82.4 |
| 037_scissors | 38.7 | 70.3 | **71.4** | 51.1 | 53.5 | 53.9 | 61.0 | 66.3 |
| 040_large_marker | 67.1 | 80.4 | 80.6 | 58.8 | 60.2 | 64.6 | 62.4 | **80.9** |
| 051_large_clamp* | 59.3 | 84.1 | 89.5 | 85.3 | 86.8 | 89.2 | **90.8** | 90.7 |
| 052_extra_large_clamp* | 44.3 | 90.3 | **91.8** | 78.9 | 85.5 | 87.8 | 89.7 | 91.3 |
| 061_foam_brick* | 92.6 | **95.5** | 87.4 | 83.3 | 84.1 | 87.7 | 86.8 | 92.9 |
| Avg. | 64.4 | **85.9** | 85.0 | 77.9 | 80.3 | 81.8 | 83.2 | **85.9** |

**TABLE 10.** Average AUC ADD scores [%] (max. th. 10 cm) achieved on the YCB-Video dataset by our algorithm, algorithms estimating object pose in single frames, and algorithms for object pose tracking.

| | PoseCNN | DOPE | [21] | [37] | GDR-Net | PoseRBPF | DeepIM | Our VDC-fuzzyTOPSIS 128 |
|---|---|---|---|---|---|---|---|---|
| Pose estimation | 61.3 | 65.8 | 72.8 | 61.0 | 84.4 | - | 81.9 | 82.2 |
| Pose tracking | - | - | - | - | - | 64.4 | **85.9** | **85.9** |

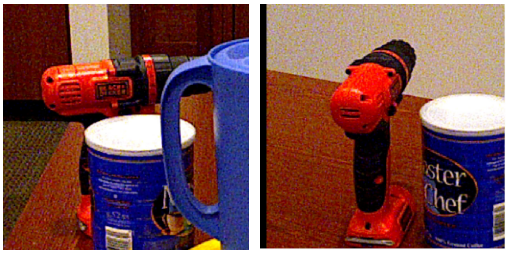

**FIGURE 7.** Appearance of the subject of interest in the first and last frames of sequence #56.

the largest gains in ADD scores. When the percentage of occlusion is quite small (c.f. image number 1000 – 1200), the TOPSIS performs similarly to the fuzzy TOPSIS.

Afterwards, we simulated situations in which the location of an object keypoint is determined with a larger error. Very often keypoints that are located on occluded/invisible part of the object are far away from the true locations of keypoints. In order to simulate scenarios with occluded/invisible object keypoints we randomly selected a single keypoint that has then been randomly translated by a vector with values sampled from (5…15) pixels. In the experiment with the moved keypoint on the power drill object the ADD score achieved by the PnP-based algorithm was smaller by almost 3.5% in comparison to ADD score of PnP-based algorithm with no keypoint movement, and it was smaller about 6.4% for fuzzy TOPSIS. The fuzzy TOPSIS was capable of detecting this randomly selected and then translated

**TABLE 11. MSE and Dice scores achieved by our network for joint object segmentation and fiducial keypoints estimation.**

| | MSE (keypoints) | Dice sc. (obj. seg.) | Dice sc. (occlud. obj. seg.) |
|---|---|---|---|
| 002_master_chef_can | 3.65 | 0.96 | 0.88 |
| 003_cracker_box | 1.83 | 0.94 | 0.81 |
| 004_sugar_box | 3.96 | 0.97 | 1.00 (no occlusions) |
| 005_tomato_soup_can | 6.11 | 0.96 | 0.93 |
| 006_mustard_bottle | 1.95 | 0.95 | 1.00 (no occlusions) |
| 007_tuna_fish_can | 1.96 | 0.96 | 0.46 |
| 008_pudding_box | 6.56 | 0.93 | 0.91 |
| 009_gelatin_box | 68.6 | 0.97 | 1.00 (no occlusions) |
| 010_potted_meat_can | 6.19 | 0.96 | 0.92 |
| 011_banana | 3.39 | 0.95 | 0.83 |
| 019_pitcher_base | 3.76 | 0.97 | 0.74 |
| 021_bleach_cleanser | 2.47 | 0.97 | 0.94 |
| 024_bowl | 2.25 | 0.97 | 0.92 |
| 025_mug | 1.54 | 0.96 | 0.85 |
| 035_power_drill | 1.49 | 0.94 | 0.81 |
| 036_wood_block | 93.2 | 0.97 | 1.00 (no occlusions) |
| 037_scissors | 64.9 | 0.93 | 0.88 |
| 040_large_marker | 2.90 | 0.92 | 0.42 |
| 051_large_clamp | 1.92 | 0.95 | 0.33 |
| 052_extra_large_clamp | 3.57 | 0.93 | 0.88 |
| 061_foam_brick | 47.7 | 0.96 | 0.89 |

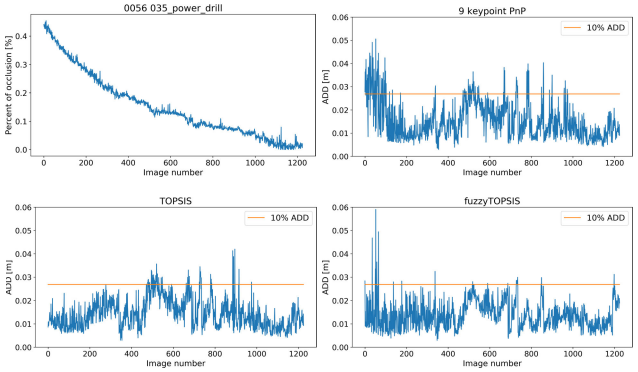


**FIGURE 8. Occlusion degree of power drill in sequence #56 vs. frame number, ADD score vs. frame number achieved by PnP, ADD score vs. frame number achieved by PnP and TOPSIS, ADD score vs. frame number achieved by PnP and fuzzy TOPSIS.**

keypoint in almost 79% of cases. Tab. 12 summarizes experimental results that have been achieved on images with the power drill object. As we can observe, the drop in the pose estimation performance for our VDC-fuzzyTOPSIS algorithm in discussed experiment is far smaller than drop in the performance for multi-criteria decision making with a single method, i.e. using only TOPSIS. The experimental results demonstrate that fuzzy TOPSIS-based algorithm holds some potential as it is capable of decreasing errors in 6D object pose tracking. The proposed hourglass-based neural network, which jointly detects nine object keypoints, predicts the object's non-occluded part, and also predicts the object's occluded part delivers valuable decision variables. The system for 6D object pose tracking has been implemented in Python with the support of Keras-GPU framework. It runs on an ordinary PC with a CPU/GPU. On a PC equipped with AMD Ryzen 7 2700, GeForce 2060 GPU the tracking was performed with nine frames per second.

**TABLE 12. ADD scores [%] achieved on toy data.**

| Toy data | ADD |
|---|---|
| PnP | 90.3 |
| TOPSIS | 95.8 |
| fuzzyTOPSIS | 98.4 |
| VDC-fuzzyTOPSIS | 98.4 |
| **Toy data with moved keypoint** | **ADD** |
| PnP | 86.5 |
| fuzzyTOPSIS | 92.0 |
| VDC-fuzzyTOPSIS | 97.8 |

## IV. CONCLUSION

Almost all SOTA algorithms both for 6D object estimation and 6D object tracking deliver the result on the basis of a single hypothesis. Different from other methods for object pose estimation, which are based on sparse object keypoints, the proposed algorithm that takes full advantage of multi-criteria decision making under uncertainties is capable of improving PnP-RANSAC pose estimates. We demonstrated experimentally that TOPSIS algorithms operating on features that are frequently utilized in such algorithms for pose estimation/tracking, including segmented masks, sparse object keypoints, and distance transform, permit achieving tangible performance gains. Experimental results demonstrated that fuzzy TOPSIS is capable of achieving far better results in comparison to TOPSIS operating on crisp values. In the ablation study, we identified components of the proposed algorithm that contribute most to performance improvements, particularly on uncertain and noisy visual data.

## REFERENCES

[1] S. M. Baas and H. Kwakernaak, "Rating and ranking of multiple-aspect alternatives using fuzzy sets," *Automatica*, vol. 13, no. 1, pp. 47–58, Jan. 1977.

[2] A. Bookstein, S. T. Klein, and T. Raita, "Fuzzy Hamming distance: A new dissimilarity measure," in *Combinatorial Pattern Matching*. Cham, Switzerland: Springer, 2001, pp. 86–97.

[3] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision in C++ With the OpenCV Library*, 2nd ed. New York, NY, USA: O'Reilly Media, 2013.

[4] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, pp. 41–75, Dec. 1997.

[5] C.-T. Chen, "Extensions of the TOPSIS for group decision-making under fuzzy environment," *Fuzzy Sets Syst.*, vol. 114, no. 1, pp. 1–9, Aug. 2000.

[6] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "PoseRBPF: A Rao–Blackwellized particle filter for 6D object pose tracking," in *Proc. Robot., Sci. Syst. (RSS)*, 2019.

[7] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van de Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[8] F. Ecer, "An extended MAIRCA method using intuitionistic fuzzy sets for coronavirus vaccine selection in the age of COVID-19," *Neural Comput. Appl.*, vol. 34, no. 7, pp. 5603–5623, Apr. 2022.

[9] I. Emovon and W. O. Aibuedefe, "FUZZY TOPSIS application in materials analysis for economic production of cashew juice extractor," *Fuzzy Inf. Eng.*, vol. 12, no. 1, pp. 1–18, Jan. 2020.

[10] Z. Fan, Y. Zhu, Y. He, Q. Sun, H. Liu, and J. He, "Deep learning on monocular object pose detection and tracking: A comprehensive overview," *ACM Comput. Surv.*, vol. 55, no. 4, pp. 1–40, Apr. 2023.

[11] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Computer Vision—ACCV 2012*. Cham, Switzerland: Springer, 2013, pp. 548–562.
[12] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, "Segmentation-driven 6D object pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3380–3389.
[13] C.-L. Hwang, Y.-J. Lai, and T.-Y. Liu, "A new approach for multiple objective decision making," *Comput. Oper. Res.*, vol. 20, no. 8, pp. 889–899, 1993.
[14] W. Jian-Giang and W. Run-Qi, "Hybrid random multi-criteria decision-making approach with incomplete certain information," in *Proc. Chin. Control Decis. Conf.*, Jul. 2008, pp. 1444–1448.
[15] A. Kaufmann and M. M. Gupta, *Introduction to Fuzzy Arithmetic: Theory and Applications* (Electrical/Computer Science and Engineering Series). New York, NY, USA: Van Nostrand Reinhold Company, 1985.
[16] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep iterative matching for 6D pose estimation," *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 657–678, Mar. 2020.
[17] Z. Li, G. Wang, and X. Ji, "CDPN: Coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7677–7686.
[18] M. Majcher and B. Kwolek, "Multiple-criteria-based object pose tracking in RGB videos," in *Proc. 14th Int. Conf. Comp. Collective Intell.*, vol. 13501. Cham, Switzerland: Springer, 2022, pp. 477–490.
[19] G. Marullo, L. Tanzi, P. Piazzolla, and E. Vezzetti, "6D object position estimation from 2D images: A literature review," *Multimedia Tools Appl.*, vol. 2022, pp. 24605–24643, Nov. 2022.
[20] T. Xu and W. Takano, "Graph stacked hourglass networks for 3D human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Cham, Switzerland: Springer, Jun. 2021, pp. 483–499.
[21] M. Oberweger, M. Rad, and V. Lepetit, "Making deep heatmaps robust to partial occlusions for 3D object pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Lecture Notes in Computer Science, vol. 11219. Cham, Switzerland: Springer, 2018, pp. 125–141.
[22] K. Park, T. Patten, and M. Vincze, "Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7667–7676.
[23] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-DoF object pose from semantic keypoints," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2011–2018.
[24] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4556–4565.
[25] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3848–3856.
[26] C. Sahin, G. Garcia-Hernando, J. Sock, and T.-K. Kim, "A review on object pose recovery: From 3D bounding box detectors to full 6D pose estimators," *Image Vis. Comput.*, vol. 96, Apr. 2020, Art. no. 103898.
[27] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 292–301.
[28] Z. Terzopoulou and U. Endriss, "The Borda class: An axiomatic study of the Borda rule on top-truncated preferences," *J. Math. Econ.*, vol. 92, pp. 31–40, Jan. 2021.
[29] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," in *Proc. CoRL*, vol. 87, 2018, pp. 306–316.
[30] R. Verma, "Fuzzy MABAC method based on new exponential fuzzy information measures," *Soft Comput.*, vol. 25, no. 14, pp. 9575–9589, Jul. 2021.
[31] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 16606–16616.
[32] H. Wang, X. Lu, Y. Du, C. Zhang, R. Sadiq, and Y. Deng, "Fault tree analysis based on TOPSIS and triangular fuzzy number," *Int. J. Syst. Assurance Eng. Manage.*, vol. 8, no. 4, pp. 2064–2070, 2017.
[33] J. Wu, B. Zhou, R. Russell, V. Kee, S. Wagner, M. Hebert, A. Torralba, and D. M. S. Johnson, "Real-time object pose estimation with pose interpreter networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 6798–6805.
[34] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Mar. 2018.
[35] X. Xie, B. L. Bhatnagar, and G. Pons-Moll, "Visibility aware human-object interaction tracking from single RGB camera," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 4757–4768.
[36] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6D pose object detector and refiner," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1941–1950.
[37] M. Zappel, S. Bultmann, and S. Behnke, "6D object pose estimation using keypoints and part affinity fields," in *Proc. RoboCup*. Cham, Switzerland: Springer, 2022, pp. 78–90.
[38] S. Zhang, W. Zhao, Z. Guan, X. Peng, and J. Peng, "Keypoint-graph-driven learning framework for object pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1065–1073.
[39] W. Zhu, H. Feng, Y. Yi, and M. Zhang, "FCR-TrackNet: Towards high-performance 6D pose tracking with multi-level features fusion and joint classification-regression," *Image Vis. Comput.*, vol. 135, Jul. 2023, Art. no. 104698.
[40] H.-J. Zimmermann, *Fuzzy Set Theory and Its Applications*, 3rd ed. Cham, Switzerland: Springer, 2001.

**MATEUSZ MAJCHER** received the M.Sc. degree in computer science from the AGH University of Science and Technology, in 2019, where he is currently pursuing the Ph.D. degree with the Institute of Computer Science. He is also a Teaching Assistant with the Institute of Computer Science, AGH University of Science and Technology. His research interests include computer vision and machine learning, particularly in the areas of object pose tracking and pose estimation.

**BOGDAN KWOLEK** received the M.Sc. degree from the Rzeszow University of Technology and the Ph.D. degree from the AGH University of Science and Technology, Kraków. He was awarded DAAD Scholarships to Bielefeld University and Technische Universitaet Muenchen, a scholarship from the French Government to INRIA, JSPS Fellowship to the Nagoya Institute of Technology, and a scholarship from Polish Government to Stanford University. He is currently a Full Professor in computer science with the AGH University of Science and Technology. His research interests include human–machine communication and deep learning. He is the author of more than 100 articles in the above-mentioned field.