Development of an Object Matcher Using Interest Points and Coherent Point Drift for a CPU-Only Robot System

Abstract

In this study, a CPU-based object matching system using C_CPD and K_CPD was evaluated in terms of execution time and accuracy. C_CPD demonstrated higher accuracy (64%) but longer execution time, while K_CPD exhibited shorter execution time but relatively lower accuracy (36%). The higher accuracy of C_CPD can be attributed to the shape representation capability of Convex Hull, whereas the faster execution speed of K_CPD is due to the characteristics of the ORB keypoint extraction method. Considering the trade-off between execution time and accuracy, C_CPD prioritizes accuracy, while K_CPD prioritizes execution speed.

keywords: K-means clustering, Convex hull, Keypoints, EM algorithm, coherent point drift, absolute orientation

Problem to solve

Object matching using deep learning requires training, which incurs significant costs in terms of time, data, and GPU resources. These requirements pose challenges for researchers and practitioners, limiting the accessibility and scalability of object matching systems. To address this problem, our research focuses on developing a CPU-only non-training object matching system.

By creating a CPU-only solution, we aim to overcome the limitations imposed by deep learning training. Our system eliminates the need for specialized hardware and extensive datasets, reducing costs and increasing accessibility. It provides an efficient and cost-effective alternative for object matching without compromising accuracy or performance.

Idea

Our proposed idea involves using either ORB keypoints combined with K means clustering or Convex Hull combined with K means clustering for feature extraction. These features are then utilized in the Coherent Point Drift (CPD) algorithm for object matching. ORB keypoints offer efficiency and robustness, while Convex Hull provides shape-based features. Our aim is to develop a CPU-only object matching system without the need for deep learning training.

Background

Interest Points - ORB

ORB features, such as Oriented FAST and Rotated BRIEF, are popular in computer vision for their computational efficiency and robustness in detecting and describing keypoints. They offer real-time performance and are particularly effective in detecting corner-like structures. However, the binary nature of the descriptors may limit their discriminative power, and they may struggle with changes in illumination. Nonetheless, ORB features strike a balance between efficiency and robustness, making them widely used for object matching in real-time applications.

Convex Hull

Convex Hull is a powerful geometric feature that efficiently captures an object's shape characteristics. It defines the smallest convex polygon enclosing the object, offering insights into its boundaries, orientation, and structure. Convex Hull is robust against outliers and has proven effective in various computer vision applications. Its simplicity, efficiency, and discriminative power make it a valuable addition to object matching algorithms, particularly in shape-based matching tasks.

Coherent Point Drift

CPD (Coherent Point Drift) aligns two sets of points by treating them as a Gaussian Mixture Model (GMM) and updating the GMM centers iteratively to maximize likelihood. It infers the rigid transformation between the sets and utilizes a closed-form solution derived from EM algorithm.

N: Number of points in the target.

x_i $(i = 1, 2, ..., N)$: Elements of the target., $x_i \in R^{2\times1}$

$X = (x_1, ..., x_N)^T$: target (data points), $X \in R^{N\times2}$

t: As the optimization progresses (with increasing time steps), the source points move according to the rigid body transformation.

$s(t)$: scaling factor, $s(t) \in R$

$R(t)$: rotation matrix, $R(t) \in R^{2\times2}$

$p(t)$: translation matrix, $p(t) \in R^{2\times1}$

$\sigma^2(t)$: isotropic covariances, $\sigma^2(t) \in R$

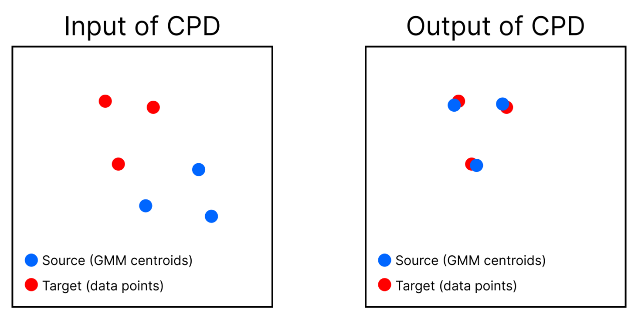
M: Number of points in the source.

$y_i(t)$ $(i = 1, 2, ..., M)$: Elements of the source., $y_i(t) \in R^{2\times1}$, $y_i(t)$ satisfies the following equation:

$y_i(t + 1) = s(t)R(t)y_i(t) + p(t)$

$Y(t) = (y_1(t), ..., y_M(t))^T$: source (GMM centroids), $Y(t) \in R^{M\times2}$

The basic structure of the problem is as described above. The key objective is to find the optimal values of s^*, R^*, p^*, σ^* using the EM algorithm. Lemma 1 is utilized during the process of optimizing the objective function with the EM algorithm.



Example of CPD Algorithm for Rigid Point Set Registration

lemma 1 Let $R(\in R^{2\times2})$ represent an unknown rotation transformation, and A$(\in R^{2\times2})$ denote a given matrix. Suppose A = $A = U\Sigma V^T$, where Σ is a diagonal matrix in descending order, resulting from the singular value decomposition. In this case, the rotation R that maximizes $tr(A^T R)$ is UCV^T, where $C = diag(1, 1, ..., 1, det(UV^T))$.

The M points in the source represent the centroids of a Gaussian Mixture Model (GMM). Each point, $y_m(t)$, in the source is generated from an independent Gaussian distribution.

$$p^t(x|m) = \frac{1}{2\pi\sigma^2(t)} exp(-\frac{|x - y_m(t)|^2}{2\sigma^2(t)})$$

The probability of generating a point, x, from the source can be expressed as a linear combination of the priors of each Gaussian distribution and the Gaussian distributions themselves. In addition to the M Gaussian distributions generated by the source, a uniform distribution noise is added to form the following equation.

$$p^t(x) = \sum_{m=1}^{M+1} p(m)p^t(m)$$

For m in the range $[1, M]$, the Gaussian distribution generated for each m has a prior of $p(m) = \frac{1-w}{M}$, and the probability density function $p^t(x|m)$ is given by $\frac{1}{2\pi\sigma^2(t)} exp(-\frac{|x - y_m(t)|^2}{2\sigma^2(t)})$. When $m = M + 1$, the prior is $p(M + 1) = w$, and the pdf $p^t(x|M + 1)$ is a constant $\frac{1}{N}$ distribution.

By introducing the EM algorithm, we can use the expected value of the complete negative log-likelihood as the objective function Q and solve it as a minimization problem. The E-step and M-step are then designed to minimize Q, allowing us to derive the CPD algorithm.

$$Q(t + 1) = -\sum_{n=1}^{N}\sum_{m=1}^{M+1} p^t(m|x_n) log(p^{t+1}(m)p^{t+1}(m))$$

The soft membership $p^t(m|x_n)$ can be derived using Bayes' theorem. $p^t(m|x_n)$ is a function that includes the already optimized parameters at time step t, viewed from the perspective of $Q(t + 1)$. Therefore, at time step $t + 1$, $p^t(m|x_n)$ can be treated as a constant. Considering the rigid body transformation of the source points y_m, we can obtain the following result in the E-step:

$$for\ m = 1, \dots M$$

$$p^t(m|x_n) = \frac{p^t(m)p^t(x_n|m)}{p^t(x_n)} = \frac{p^t(m)p^t(x_n|m)}{\sum_{k=1}^{M+1} p^t(k)p^t(k)}$$

$$= \frac{\frac{1-w}{M}\frac{1}{2\pi\sigma^2(t)} exp(-\frac{|x_n - (s(t)R(t)y_m(t) + p(t))|^2}{2\sigma^2(t)})}{\frac{w}{N} + \sum_{k=1}^{M}\frac{1-w}{M}\frac{1}{2\pi\sigma^2(t)} exp(-\frac{|x_n - (s(t)R(t)y_k(t) + p(t))|^2}{2\sigma^2(t)})}$$

$$= \frac{exp(-\frac{|x_n - (s(t)R(t)y_m(t) + p(t))|^2}{2\sigma^2(t)})}{2\pi\sigma^2(t)\frac{w}{1-w}\frac{N}{M} + \sum_{k=1}^{M} exp(-\frac{|x_n - (s(t)R(t)y_k(t) + p(t))|^2}{2\sigma^2(t)})}$$

$$for\ m = M + 1$$

$$p^t(m|x_n) = \frac{2\pi\sigma^2(t)\frac{w}{1-w}\frac{N}{M}}{2\pi\sigma^2(t)\frac{w}{1-w}\frac{N}{M} + \sum_{k=1}^{M} exp(-\frac{|x_n - (s(t)R(t)y_k(t) + p(t))|^2}{2\sigma^2(t)})}$$

If we eliminate all terms unrelated to $s(t + 1)$, $R(t + 1)$, $p(t + 1)$ and $\sigma(t + 1)$ from $Q(t + 1)$, the resulting expression is as follows:

$$Q(t + 1) = \frac{1}{2\sigma^2(t+1)}\sum_{n=1}^{N}\sum_{m=1}^{M} p^t(m|x_n)$$
$$(x_n - [s(t + 1)R(t + 1)y_m(t + 1) + p(t + 1)])^2$$
$$+ N_p(t)log(\sigma^2(t + 1)),\ N_p(t) \equiv \sum_{n=1}^{N}\sum_{m=1}^{M} p^t(m|x_n)$$

In order to obtain the results of the M-step, we need to solve the following optimization problem:

Minimize $Q(t;\ s,\ R,\ p,\ \sigma)$
subject to

$R(t)R(t)^T = R(t)^T R(t) = I,$
$det(R(t)) = 1$

If we differentiate Q with respect to $p(t)$ and set it to zero, we can find the optimal solution for pt satisfying the first-order necessary condition (FONC). Let $P(t)$ be an M×N matrix defined as $P(t)_{mn} = p^t\left(m|x_n\right)$, where $p(t)$ is an element of the matrix. By utilizing the column vector $1_{dimension\ of\ rows}$ with the appropriate dimensions in the computations, we can express the optimal solution for pt as follows:

$$p^*(t) = \mu_x(t) - s(t)R(t)\mu_y(t)$$
$$\mu_x(t) \equiv \frac{1}{N_p(t-1)} X^T P^T(t-1)1_M,\ \mu_x(t) \in R^{2\times1}$$
$$\mu_y(t) \equiv \frac{1}{N_p(t-1)} Y^T(t)P(t-1)1_N,\ \mu_x(t) \in R^{2\times1}$$

By substituting $p^*(t)$ into $Q(t)$, we can simplify the expressions for $\mu_x(t)$ and $\mu_y(t)$.

$$Q(t) = \frac{1}{2\sigma^2(t)} [tr(\hat{X}^T(t)\, diag(P^T(t-1)1_M)\, \hat{X}(t))$$
$$-\ 2s(t)tr(\hat{X}^T(t)\, P^T(t-1)\, \hat{Y}(t)R^T(t))$$
$$+\ s^2(t)tr(\hat{Y}^T(t)\, diag(P(t-1)1_N)\, \hat{Y}(t))]$$
$$+\ N_p(t-1)log(\sigma^2(t))$$
$$\hat{X}(t) \equiv X - 1_N\mu_x^T(t),\ \hat{Y}(t) \equiv Y(t) - 1_M\mu_y^T(t)$$

To obtain the optimal $R(t)$, we only need to observe the terms related to $tr(\hat{X}^T(t)\, P^T(t-1)\, \hat{Y}(t)R^T(t))$ in $Q(t)$. Additionally, by utilizing properties such as the transpose of the trace and the invariance of cyclic matrix permutations, we can simplify $tr(\hat{X}^T(t)\, P^T(t-1)\, \hat{Y}(t)R^T(t))$ to $tr((\hat{X}^T(t)\, P^T(t-1)\, \hat{Y}(t))^T R(t))$.

Maximize $tr(A^T R(t)),\ A \equiv \hat{X}^T(t)\, P^T(t-1)\, \hat{Y}(t)$ subject to
$R(t)R(t)^T = R(t)^T R(t) = I,$
$det(R(t)) = 1$

$$R^*(t) = UCV^T$$
$$U\Sigma V^T \equiv svd(\hat{X}^T(t)\, P^T(t-1)\, \hat{Y}(t)),$$
$$C \equiv diag(1, \dots, 1, det(UV^T))$$

Based on the provided information, we can reconstruct $Q(t)$ and find the optimal solution for $s(t)$ and $\sigma(t)$ that satisfies the FONC (first-order necessary condition). Here is the expression:

$$s^*(t) = \frac{tr(A^T R(t))}{tr(\hat{Y}^T(t)\, diag(P(t-1)1_N)\, \hat{Y}(t))}$$
$$(\sigma^*)^2(t) = \frac{1}{2N_p(t-1)} (\hat{X}^T(t)\, diag(P^T(t-1)1_M)\, \hat{X}(t)$$
$$-\ s(t)tr(A^T R(t)))$$

Here is the combined formulation of the E-step and M-step in algorithmic form, with M-step performed in the order of R^*, s^*, p^* and σ^*:

Algorithm 1 Coherent Point Drift for Rigid Point Set Registration
// initialization
$R \leftarrow I$
$p \leftarrow 0$
$s \leftarrow 0$
$\sigma^2 \leftarrow \frac{1}{2NM} \sum_{n=1}^{N} \sum_{m=1}^{M} (x_n - y_m)^2$
$w \in [0, 1]$
$compute\ Q_{before}$
$while\ True$

// move GMM centroids

$for\ m \in [1, M]$

$$y_m \leftarrow sRy_m + p$$

// E-step

$for\ m \in [1, M],\ n \in [1, N]$

$$P_{mn} \leftarrow \frac{exp(-\frac{|x_n - y_m|^2}{2\sigma^2})}{2\pi\sigma^2 \frac{w}{1-w}\frac{N}{M} + \sum_{k=1}^{M} exp(-\frac{|x_n - y_k|^2}{2\sigma^2})}$$

// M-step

$$N_p \leftarrow 1_M^T P 1_N$$

$$\mu_x \leftarrow \frac{1}{N_p} X^T P^T 1_M$$

$$\mu_y \leftarrow \frac{1}{N_p} Y^T P 1_N$$

$$\hat{X} \leftarrow X - 1_N \mu_x^T$$

$$\hat{Y} \leftarrow Y - 1_M \mu_y^T$$

$$A \leftarrow \hat{X}^T P^T \hat{Y}$$

$$U,\ \Sigma,\ V \leftarrow svd(A)$$

$$C \leftarrow diag(1, \ldots, 1, det(UV^T))$$

$$R \leftarrow UCV^T$$

$$s \leftarrow \frac{tr(A^T R)}{tr(\hat{Y}^T\ diag(P 1_N)\ \hat{Y})}$$

$$p \leftarrow \mu_x - sR\mu_y$$

$$\sigma^2 \leftarrow \frac{1}{2N_p}\left(\hat{X}^T diag\left(P^T 1_M\right)\hat{X} - s\, tr\left(A^T R\right)\right)$$

$compute\ Q_{after}$

// Converge condition

$If\ |Q_{after} - Q_{before}| < \epsilon$

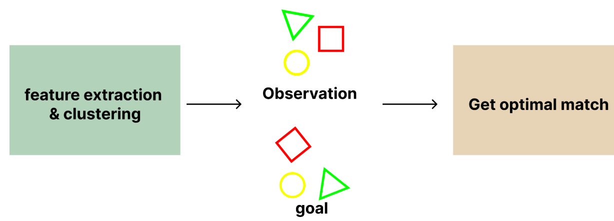
Break

$Q_{before} \leftarrow Q_{after}$

$return\ R,\ s,\ p$

Implementation

We divided our development process into frontend and backend. The frontend involves extracting keypoints using ORB and convex hull extraction using the Graham scan algorithm. In the backend, we consistently employed the CPD algorithm, adapting to different input types, regardless of the frontend method used.



(1) convex hull + CPD

To obtain a point cloud of the objects in the workspace when given the goal and observation images in image format, you can follow these steps:

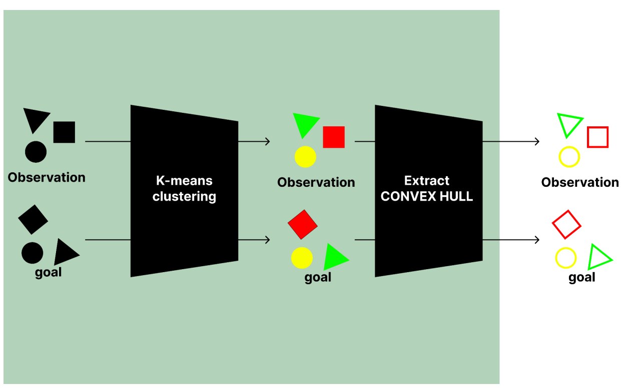
A. Perform K-means clustering: Apply K-means clustering to the goal and observation images to generate a point cloud with a number of clusters equal to the number of objects. This process groups similar pixels together, resulting in distinct clusters representing different objects.

B. Preserve cluster information: Maintain the cluster information obtained from K-means clustering for each pixel in the images. This information will be used to differentiate between clusters and associate keypoints with specific objects.

C. Apply the Graham scan algorithm: Utilize the Graham scan algorithm for each cluster/object to extract keypoints. The Graham scan algorithm identifies the convex hull of the points within each cluster, capturing the outer boundary of the object.

D. Extract keypoints: Extract keypoints from each convex hull using a desired feature extraction method, such as ORB. These keypoints represent distinctive points on each object.

E. Preserve object identification: Maintain the association between keypoints and their corresponding objects by utilizing the cluster information obtained from K-means clustering. This information helps distinguish keypoints belonging to different objects.



[convex hull extraction process using the Graham scan algorithm]

When given a set of clusters in the form of convex hulls as input, the following steps can be taken to find the best match among all possible pairwise combinations of objects in the goal and observation:

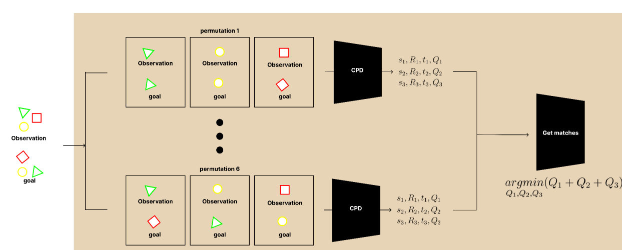
A. Generate all pairwise combinations: Generate all possible pairwise combinations of objects between the goal and observation images. This will result in a set of pairs representing potential matches.

B. Apply CPD algorithm on object-level: Pass each pair of objects through the CPD algorithm, treating them as individual point clouds. The CPD algorithm estimates the transformation parameters and computes the quality measure Q for each pair.

C. Calculate Q sum: Calculate the sum of the Q values obtained from the CPD algorithm for each pair of objects. This represents the overall quality of the match between the objects.

D. Find the minimum Q sum: Identify the pair of objects that yields the minimum Q sum. This corresponds to the best matching pair of objects.

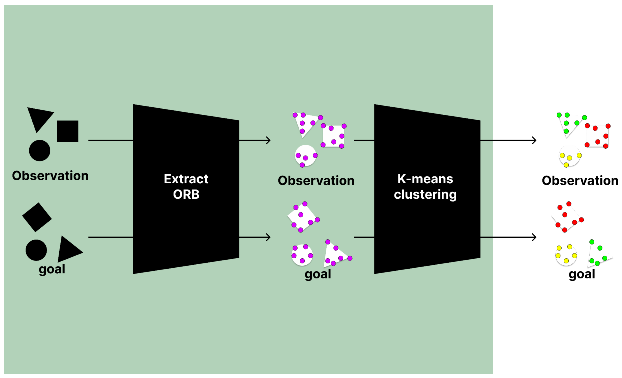
By following these steps, you can generate all possible pairwise combinations of objects, apply the CPD algorithm to estimate transformations and compute Q values, and ultimately find the pair of objects that results in the smallest Q sum. This approach allows for the identification of the best match among all possible matches between objects in the goal and observation images.



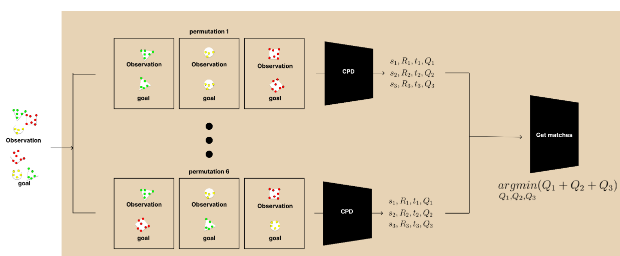
[optimal match using the Coherent Point Drift (CPD) algorithm based on the convex hull]

(2) ORB + CPD

Both the ORB + CPD and convex hull + CPD methods share a similar approach with the main difference being the use of ORB for feature extraction in the former. In both methods, keypoints are extracted in the frontend, and the CPD algorithm is employed in the backend for object matching. The overall process remains the same, focusing on keypoint extraction and utilizing CPD for matching.



[keypoints extraction using ORB]

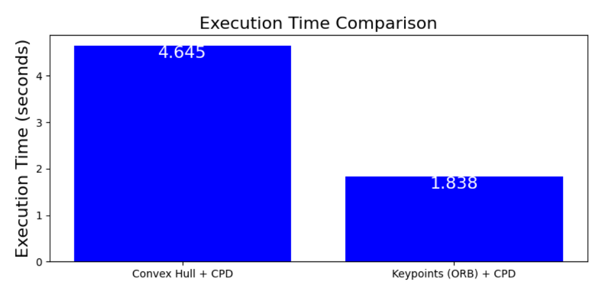


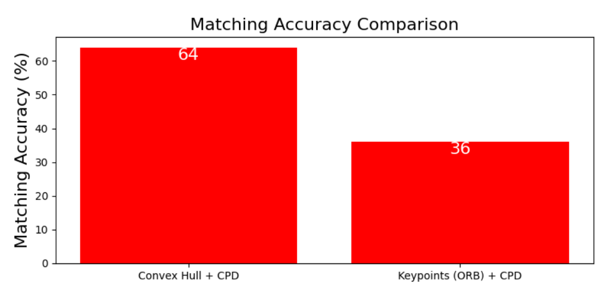
[optimal match using the Coherent Point Drift (CPD) algorithm based on the keypoints]

Evaluation Result

The experiment results showed that Convex Hull + CPD (C_CPD) and Keypoints + CPD (K_CPD) were evaluated in terms of execution time and object matching accuracy. The experiment was conducted on three objects, and if any object failed to match the ground truth, it was considered a "fail." The execution time for C_CPD was measured to be 4.6 seconds, while K_CPD took 1.8 seconds, indicating that K_CPD was faster. However, in terms of accuracy, C_CPD achieved 64%, outperforming K_CPD, which achieved 36%.

The longer execution time of C_CPD compared to K_CPD can be attributed to the additional computations required by the Convex Hull–based algorithm. Convex Hull involves complex calculations to accurately capture the object's outer boundaries, necessitating more computational resources and time. However, the higher accuracy of C_CPD can be attributed to the ability of Convex Hull to effectively represent the overall shape of the object. Convex Hull provides a simple and concise representation of the object's outer boundaries and exhibits robustness against noise and outliers. Therefore, C_CPD, utilizing Convex Hull as a feature, can better capture the shape and contours of the object, resulting in more accurate matching.





The faster speed of K_CPD can be attributed to the characteristics of the ORB keypoint extraction method. ORB is known for its computational efficiency and robustness in keypoint detection and description. By utilizing ORB as the basis, K_CPD reduces the execution time. However, the binary nature of ORB descriptors may result in relatively lower discriminative power compared to other descriptors, and it may have limitations in handling changes in lighting conditions. Consequently, K_CPD may exhibit relatively lower accuracy compared to C_CPD.

Considering the trade-off between execution time and accuracy, C_CPD provides higher accuracy at the cost of longer execution time, while K_CPD offers faster execution but lower accuracy.

Contributions

In this paper, we propose a novel approach to object matching using Q (EM objective) Simulation Optimization of Coherent Point Drift (CPD). Our approach involves the utilization of various feature extractors, namely Convex Hull and ORB, as part of the frontend for feature extraction.

To evaluate the performance of our approach, we created a test dataset by capturing images of three different objects. The dataset was carefully designed by shooting and randomly placing the objects to ensure diversity and complexity in the matching scenarios.

We conducted a comprehensive measurement and comparison of the object matcher's performance, focusing on accuracy and execution time. By analyzing the results, we assessed the effectiveness and efficiency of our proposed approach in achieving accurate object matching.

The findings of our study provide valuable insights into the capabilities and limitations of the Q (EM objective) Simulation Optimization of CPD approach for object matching. The performance evaluation sheds light on the strengths and weaknesses of the different feature extractors, Convex Hull and ORB, and their impact on the overall matching accuracy and execution time.

Overall, this research contributes to the advancement of object matching techniques by introducing a new approach and providing empirical evidence of its performance. The results serve as a benchmark for future developments and optimizations in the field of object matching using Q (EM objective) Simulation Optimization of CPD.

Discussion

The study suggests several avenues for future research in object matching. Alternative keypoint extractors, such as SIFT or SURF, should be explored to potentially improve matching performance. Incorporating 3D position information and leveraging richer data inputs, like RGB-D or multi-channel information, could enhance the CPD algorithm's understanding of object spatial characteristics. Developing a more sophisticated CPD algorithm by incorporating color or texture-based models would improve matching capabilities in diverse object appearances and challenging visual conditions. These advancements have the potential to expand the applicability and enhance the performance of object matching systems in computer vision.

References

Moshayedi, Ata Jahangir, et al. "Deep Learning Application Pros And Cons Over Algorithm Deep Learning Application Pros And Cons Over Algorithm." *EAI Endorsed Transactions on AI and Robotics* 1.1 (2022).

Osuna, Edgar, and Osberth De Castro. "Convex hull in feature space for support vector machines." *Advances in Artificial Intelligence—IBERAMIA 2002: 8th Ibero-American Conference on AI Seville, Spain, November 12–15, 2002 Proceedings 8*. Springer Berlin Heidelberg, 2002.

Zhuo, Li, et al. "ORB feature based web pornographic image recognition." *Neurocomputing* 173 (2016): 511-517.

Myronenko, A., & Geng, X. (2010). "Point Set Registration: Coherent Point Drift." IEEE Transactions on Pattern Analysis and Machine Intelligence, (TPAMI).