Observation-Based Inverse Kinematics for Visual Servo Control

Daniel Nikovski[a]
Mitsubishi Electric Research Labs, Massachusetts, U.S.A.

Keywords: Robotics, Learning Control, Configuration Learning, Visuomotor Policies.

Abstract: We propose a method for estimating the joint configuration of articulated mechanisms without joint encoders and with unknown forward kinematics, based solely on RGB-D images of the mechanism captured by a stationary camera. The method collects a sequence of such images under a suitable excitation control policy, extracts the 3D locations of keypoints in these images, and determines which of these points must belong to the same link of the mechanism by means of testing their pairwise distances and clustering them using agglomerative clustering. By computing the rigid-body transforms of all bodies with respect to the keypoints' positions in a reference image and analyzing each body's transform expressed relative to all other bodies' coordinate reference frames, the algorithm discovers which pairs of bodies must be connected by a single-degree-of-freedom joint and based on this, discovers the ordering of the bodies in the kinematic chain of the mechanism. The method can be used for pose-based visual servocontrol and other robotics tasks where inverse kinematics is needed, without providing forward kinematics or measurements of the end tool of the robot.

1 INTRODUCTION

Humans and animals possess remarkable abilities to execute very complex motions based on sensory observations, starting from hunting for prey and avoiding predators and ranging to precise motion such as hitting a fast moving ball with a bat or assembling a complex electronic device. Over the course of millions of years of evolution, one sensory modality – vision – has provided an overwhelming advantage when performing such motions, and modern control engineering and artificial intelligence (AI) have sought to emulate computationally the abilities of living organisms to interpret visual data for the purpose of directing intelligent behavior. This effort, combined with the ever decreasing cost and increasing performance of high-resolution cameras and compact embedded micro-controllers, has brought about a generational change in robotics, where rigidly-programmed robots executing identically repeated motions are gradually being replaced by robots that can interpret visual input and adjust their motion accordingly.

However, this transition is associated with the cost of manually developing an observer of the system under control, employing computer vision and control engineering techniques (Szeliski, 2022; Franklin et al., 2015). It is thus very appealing economically to automate the process of constructing observers for nonlinear systems based on visual observations. The field of state representation learning (SRL) for control addresses the general problem of producing compact descriptors of the state of a dynamical system from observation data and their use for designing control policies (Lesort et al., 2018). Methods proposed in the closely related field of nonlinear system identification also often construct state representations as their byproducts (Nelles, 2020). However, general methods in these fields, almost exclusively based on machine learning algorithms, often suffer from very high sample complexity, i.e., they need to see very many training examples in order to learn a suitable model and/or state representation, often limiting their applicability to relatively simple systems with few independent state variables. Moreover, such methods tend to produce state representations where the contributions of the individual state variables are conflated. This is in contrast with the traditionally hand-crafted perception modules in AI, where a scene descriptor is usually factored into the states of the objects the scene consists of, or state vectors in control systems engineering, which are similarly a concatenation of the states of the system's individual components.

It is thus highly desirable to be able to construct from training data useful state representations that are factored across a set of moving objects that exist in a

[a] https://orcid.org/0000-0003-2919-645X

Nikovski, D.
Observation-Based Inverse Kinematics for Visual Servo Control.
DOI: 10.5220/0013701100003982

In *Proceedings of the 22nd International Conference on Informatics in Control, Automation and Robotics (ICINCO 2025) - Volume 1*, pages 200-207
ISBN: 978-989-758-770-2; ISSN: 2184-2809

scene corresponding to a controllable system in motion. An example of such a system is an articulated robot arm consisting of several rigid bodies (links) connected to one another by means of joints. The links are arranged in a kinematic chain and the joints connecting them can be translational or rotational.

Most industrial manipulators fit this definition. Other examples are cranes (gantry, boom, beam, etc.) whose mechanism varies depending on what kind of joints are used. Automatic control of industrial manipulators has been researched over many decades (Siciliano and Khatib, 2016), and automation of cranes has also advanced in recent years (Mojallizadeh et al., 2023). Owing to their second-order dynamics, the state of such mechanisms can be expressed as the set of joint positions (angles or displacements) as well as their velocities.

Although the joint positions of many articulated mechanisms can be measured by means of encoders, in other cases, for example for the cables hoisting the load of a crane, this is not feasible. An alternative is to implement an estimator based on visual data and it is desirable to automate this process. We propose one such algorithm for configuration descriptor construction from image data. The main insight behind it is that if we can reliably detect and track 3D keypoints in sequences of such images, we can recover the mechanism's configuration much more easily than if we were working with the raw pixels in the images. Section 2 describes the problem setting and the assumptions about visual data we make, and section 3 describes some approaches to solving the problem. Section 4 describes our proposed method and Section 5 presents its empirical verification. Section 6 proposes future directions and concludes.

2 PROBLEM DEFINITION

We are interested in the accurate estimation of the configuration of an articulated mechanism, such as a robot arm, entirely from camera images, and the use of this configuration estimate for the purposes of control of the mechanism. The mechanism comprises an unknown number of rigid bodies connected in a kinematic chain via single-degree-of-freedom (DoF) joints, whose types – either prismatic or revolute – are not known *a priori*. Furthermore, the size, appearance, and ordering of the rigid bodies within the chain are also unknown, precluding direct recovery of the mechanism's true state. However, if a configuration representation can be derived from visual data that is equivalent to the true configuration – i.e., it maintains a one-to-one correspondence – then state-based control techniques can still be employed for effective control of the mechanism.

We consider a set S of N distinctive 3D keypoints tracked over T time steps, yielding measurements of the form $p_{ik} = [x_{ik}, y_{ik}, z_{ik}]^T$ for $i = 1, \ldots, N$ and $k = 1, \ldots, T$. The collection of keypoint measurements at time k is denoted by the matrix $P_k = [p_{ik}]^T \in \mathbb{R}^{N\times 3}$. These 3D positions can be estimated using an RGB-D camera that captures a sequence of T frames while the mechanism is actuated under a persistently exciting control policy. Distinctive features, such as corners, are extracted from the RGB images (Szeliski, 2022), and their 3D positions are computed in the camera frame using the corresponding depth data and the camera's intrinsic parameters. We do not assume continuous visibility of all keypoints; some may be occluded at certain time steps, resulting in missing data. Moreover, the visible keypoints are subject to measurement noise due to pixel quantization, limited depth resolution, and potential mechanical disturbances. Finally, we assume that the number of keypoints N significantly exceeds the number of rigid bodies n, ensuring that each body is associated with at least four, and preferably more, visible keypoints to enable reliable pose estimation in the camera frame.

Given the (potentially sparse) data tensor $P = [P_k] \in \mathbb{R}^{N\times T\times 3}$, where $k = 1, \ldots, T$, our objective is to infer the underlying structure and motion of the articulated mechanism. Specifically, we aim to determine: (i) the number of rigid bodies present in the scene; (ii) the assignment of each of the N keypoints to their corresponding rigid body; (iii) the pose of each body relative to a reference pose at every time step; (iv) the connectivity structure of the mechanism, identifying which bodies are linked via single-DoF joints; and (v) the joint configurations over time, expressed relative to a reference joint position. The reference body poses are defined with respect to a chosen time step, such as the initial frame in the sequence, when the reference joint positions are assumed to be zero.

At run time, we aim to bring the mechanism to a goal state specified implicitly by means of an image of the mechanism in the desired goal state, as is customary in the field of visual servocontrol (VS) (Chaumette et al., 2016). Using steps (iii) and (v) above and the mechanism structure discovered in steps (i), (ii), and (iv), we estimate the desired goal joint state. Then, we control the articulated mechanism by repeatedly (a) capturing an RGB-D image of the mechanism; (b) determining the 3D positions of the keypoints in it; (c) computing the pose of each body with respect to the reference pose, analogously to step (iii) in the analysis stage above; (d) computing the joint configuration at this time, again analogously to step (v) above; (e) computing a feedback error with respect to the desired goal configuration, in estimated joint space; and (f) computing a control signal based on the feedback error that aims to bring the error to zero, using a suitable control method, such as proportional-integral-derivative (PID) control.

3 RELATED WORK

Some deep reinforcement learning (DRL) algorithms have achieved several successes in learning visuomotor control policies directly in terms of high-dimensional observations, including camera images, but these typically require millions of exprimental trials. There do exist some DRL algorithms with acceptable sample complexity that can be trained on real robotic hardware, but not quite matching the problem we are interested in. For example, the Guided Policy Search (GPS) algorithm has been able to learn control policies for various difficult robotics tasks (Levine et al., 2016). However, this algorithm did have access to the true low-dimensional state of the system to compute optimal state-based policies at training time, later turning these policies into observation-based ones using supervised learning. In contrast, we do not assume access to the true state at any time.

Image-based visual servocontrol (IBVS) algorithms could be very effective, but usually assume that all features will be visible at all times, otherwise they would fail (Chaumette et al., 2016). Another class of VS algorithms known as pose-based visual servocontrol (PBVS) algorithms operate by first estimating the current state of the mechanism from images, essentially equivalent to estimating its configuration, followed by conventional control in joint space. However, this cannot be done analytically without a kinematic model of the mechanism and knowledge of how it would look like in different configurations. An alternative is to apply SRL to learn compact representations of the controlled system, including articulated mechanisms, and use these representations for various control and decision-making tasks, such as VS for goal-reaching behavior, reinforcement learning, and imitation learning (Lesort et al., 2018). How to perform SRL reliably and efficiently is thus a key problem in the field of robot learning.

Deep neural networks (DNNs) with a bottleneck layer can be trained to mimic their input, forcing them to compress the high-dimensional observations into a compact state descriptor. The resulting Spatial Autoencoders (SAE) and dynamic models have been used in learning control policies on real robots (Finn et al., 2016; Jonschkowski and Brock, 2015; Wahlström et al., 2015). However, this approach has turned out difficult to scale up to systems with more DoFs due to the quickly increasing sample complexity, a common issue for DNNs. It also reflects the combinatorial explosion of possible configurations as the number of joints in the mechanism increases; sampling the configuration space exhaustively is correspondingly computationally demanding.

One possible solution to handling this complexity is to leverage the decomposability of mechanisms (and natural scenes, in general) into independent objects whose pose can be estimated independently. It can be surmised that naturally intelligent humans and animals have a built-in inductive bias towards learning object-centric representations of the world, and general-purpose SRL method lack this bias. Recent work in SRL has focused on object-centered SRL, such as the Slot Attention for Video (SAVi) algorithm introduced in (Kipf et al., 2021). It processes sequences of images and extracts features essential for object tracking and representation. However, learning these features typically demands large datasets and is highly sensitive to the initial configuration of the neural network, particularly its convolutional filters.

4 OBSERVATION-BASED INVERSE KINEMATICS AND CONTROL OF MECHANISMS

The difficulties associated with DNN-based SRL methods prompt the question of whether neural networks – with their long training times, high sample complexity, and imperfect reliability – are even necessary. Instead, we propose feeding the SRL algorithm not with raw image pixels, but with the spatial coordinates of a set of corresponding keypoints across images. By analyzing the relative motion of these keypoints over time, the method statistically determines which keypoints belong to the same rigid body, estimates object poses in the camera frame, and infers the kinematic chain structure. The result is a compact representation of the mechanism's configuration, equivalent to a vector of joint positions. We can think of the mapping from images to configurations as a form of observation-based inverse kinematics (OBIK). Combined with joint-space control algorithms, this results in a new PBVS method that requires neither prior knowledge of the mechanism's kinematics, nor measurements of joint angles or end-tool position by means of sensors, thus eliminating the cost of such sensors from the cost of the entire system. The steps of the method are described below.

4.1 Assignment of Points to Rigid Bodies

The algorithm begins by identifying the number of distinct moving bodies present in the scene — including the static background – and assigning each point uniquely to one of these bodies. This is a task often performed in Structure from Motion (SfM) estimation (Szeliski, 2022). The general approach is to test whether pairs of points maintain a constant distance over time – a hallmark of rigid motion – and cluster the full set of N points into groups where each pair within a group satisfies the rigid body assumption (RBA) between them.

To implement this approach, we require a dissimilarity metric for each pair of points that captures the extent to which the Euclidean distance between them violates the RBA. A viable option for this metric is the variance of the distance D_{ij} between the two points p_i and p_j. Assuming the presence of measurement noise, this distance can be treated as a random variable. Its mean $\bar{d}_{ij}$ and sample variance s_{ij}^2 can be computed using data from the time steps during which both points are observable, as indicated by the indicator variables o_{ik}, which are equal to 1 if point i is observable at time k and 0 otherwise:

$$d_{ijk} = \begin{cases} \|p_{ik} - p_{jk}\| & \text{if } (o_{ik} = 1) \wedge (o_{jk} = 1) \\ \text{undefined} & \text{otherwise} \end{cases} \tag{1}$$

$$T_{ij} = \sum_{k=1}^{T} o_{ik} o_{jk} \qquad \bar{d}_{ij} = \frac{1}{T_{ij}} \sum_{\substack{k=1 \\ o_{ik}=1 \\ o_{jk}=1}}^{T} d_{ijk} \tag{2}$$

$$s_{ij}^2 = \frac{1}{T_{ij} - 1} \sum_{\substack{k=1 \\ o_{ik}=1 \\ o_{jk}=1}}^{T} (d_{ijk} - \bar{d}_{ij})^2 \tag{3}$$

To estimate the variances s_{ij}^2, the distance between each pair of points must be measured at least twice. Therefore, if the condition $\forall j, T_{ij} \geq 2$ is not met for a given point i, that point is excluded from the set.

However, the sample variance does not lie within a uniform interval, but can vary significantly depending on the range of motion of bodies in the scene, thus potentially confusing clustering algorithms. A more effective dissimilarity metric may be obtained by framing the problem as one of statistical hypothesis testing. The goal is to test whether the distance between two points – treated as a random variable – is not constant, meaning its variance is significantly greater than what would be expected from measurement noise alone. A suitable method for comparing variances is the F-test. To apply this test, we need an estimate of the measurement noise variance σ_{noise}^2. This can be obtained either from the camera's specifications or experimentally, by measuring the variation of keypoints' positions in a completely static scene.

The F-test is conducted by setting up a null hypothesis H_0 that the variance $\sigma_{ij}^2 = \sigma_{noise}^2$, and an alternative hypothesis H_A that $\sigma_{ij}^2 > \sigma_{noise}^2$. This is a one-sided F-test, as the true variance σ_{ij}^2 can only be equal to or greater than the noise variance σ_{noise}^2 – the latter serving as the Cramér-Rao lower bound for the former – so it is not physically meaningful for σ_{ij}^2 to be smaller. (Although the observed sample variance s_{ij}^2 may occasionally fall below σ_{noise}^2, such cases clearly indicate a constant distance.)

The F-statistic is calculated as $F_{ij} = s_{ij}^2 / \sigma_{noise}^2$, which follows an F-distribution with degrees of freedom $(T_{ij} - 1, T_{ij} - 1)$. The corresponding p-value is given by $p = \Pr(F > F_{ij}) = 1 - CDF(F_{ij})$, where $CDF(\cdot)$ denotes the cumulative distribution function of the F-distribution. This p-value represents the probability of observing such a high F-statistic under the assumption that H_0 (i.e., the points belong to the same object) is true. A low p-value (below a chosen threshold) leads to rejection of the null hypothesis in favor of the alternative H_A, suggesting that the points likely belong to different objects.

However, for our application, we do not need to select a specific confidence level. Instead, we use the complement of the p-value, $q_{ij} = 1 - p = CDF(F_{ij})$, as the dissimilarity metric between points i and j for clustering purposes. This measure typically ranges from around 0.5 (when $s_{ij}^2 \approx \sigma_{noise}^2$ and thus $F_{ij} \approx 1$) to values approaching 1 (when $F_{ij} \gg 1$). This more consistent scaling facilitates the clustering of points into distinct objects.

4.2 Clustering of Keypoints

Clustering can then be performed using a suitable algorithm that accepts a dissimilarity matrix as input. A good choice is agglomerative clustering (Murtagh and Contreras, 2012), with complete linkage, because if a subset of points correspond to the same object, the RBA must hold between each pair of them. Let S_l, for $l = 1, \hat{n}_0$, denote the resulting clusters (subsets) of points from the original set S, and let $N_l = |S_l|$ be the number of points in cluster l. Clusters with $N_l < 4$ are discarded, as they are insufficient for estimating the pose of the associated object. The number $\hat{n}$ of remaining clusters then serves as an estimate of the true number n of rigid bodies in the scene, including the static background.

4.3 Configuration Estimation

Once the points have been grouped into clusters of size at least 4, we can estimate the relative poses of the identified bodies with respect to a reference configuration, defined by the measured positions of the points at a chosen reference time frame. Let these reference positions be denoted by $p_{i0} = [x_{i0}, y_{i0}, z_{i0}]^T$, and let $P_0 = [p_{i0}]^T \in \mathbb{R}^{N\times 3}$ for $i = 1, \ldots, N$.

Define $\mathcal{P}_{l0} = [\mathrm{p}_{0i}]_{i=1}^{N_l} = [[x_{0i}, y_{0i}, z_{0i}]^T]_{i=1}^{N_l}$ as the matrix of reference positions for the points in cluster l, and $\mathcal{P}_{lk} = [\mathrm{p}_{ik}]_{i=1}^{N_l} = [[x_{ik}, y_{ik}, z_{ik}]^T]_{i=1}^{N_l}$ as the matrix of positions for the same points at time k, with each column representing a point. We can estimate the rigid body transformation (RBT) that best aligns $\mathcal{P}_{l0}$ to $\mathcal{P}_{lk}$ in the least-squares sense: $\mathrm{p}_{ik} \approx R_{lk}\mathrm{p}_{i0} + t_{lk}$, $i = 1, \ldots, N_l$, where R_{lk} is a 3×3 rotation matrix and t_{lk} is a 3×1 translation vector. This can be achieved using Procrustes superimposition via the Kabsch-Umeyama algorithm (Umeyama, 1991).

By applying this procedure to each cluster, we obtain a set of $\hat{n}$ rigid body transformations that fully describe the configuration of the mechanism. This set can serve as a compact configuration descriptor, significantly lower in dimensionality than the original RGB-D images or keypoints from which it was derived, making it suitable for tasks such as monitoring and control. However, this representation is not minimal, requiring $6\hat{n}$ numbers, whereas the true configuration is described by only $\hat{n}$ joint positions.

The computed RBTs for each identified rigid body can be further analyzed to infer the underlying kinematic structure of the mechanism and to construct a more compact configuration descriptor. When two bodies are adjacent in a kinematic chain and connected by a single-DoF joint, their relative RBT will also exhibit only one DoF – either translational or rotational. Recall that the RBTs obtained so far are expressed relative to a reference pose defined by the point set $\mathcal{P}_{l0}$, all in the inertial (camera) frame. The relative pose of object m can be expressed instead with respect to object l at time k, and denoted as ${}^{l}R_{mk}$. The relative rotation satisfies the relation $R_{mk} = R_{lk}\,{}^{l}R_{mk}$, where leading superscripts indicate the frame in which the rotation is expressed, and the absence of such superscript indicates the world (here, camera) frame, yielding ${}^{l}R_{mk} = R_{lk}^T R_{mk}$. Similarly, the relative translation is given by ${}^{l}t_{mk} = R_{lk}^T(t_{mk} - t_{lk})$.

These relative poses are instantaneous, for time step k. To determine the number of DoFs between two bodies, we can analyze how their relative pose evolves over time. If the relative position or orientation remains constant (within tolerance), it indicates zero translational or rotational DoF, respectively.

To infer translational DoFs, we examine the rank of the $3\times T$ matrix $\mathcal{T}_{lm} = [{}^{l}t_{m1}\ {}^{l}t_{m2}\ \ldots\ {}^{l}t_{mT}]$. If body m moves translationally with respect to body l along a fixed axis (as in a prismatic joint), each relative translation ${}^{l}t_{mk}$ should lie along the same line in $\mathbb{R}^3$, implying that $\mathrm{rank}(\mathcal{T}_{lm}) = 1$. This can be detected using singular value decomposition (SVD) on $\mathcal{T}_{lm}$. (It is important to note that $\mathrm{rank}(\mathcal{T}_{lm}) = 1$ also holds when the relative position is constant, but not zero (meaning zero translational DoF), so this case must be identified beforehand to avoid misinterpretation.)

Detecting a single rotational DoF is more challenging, for two reasons. First, rotation matrices do not naturally lend themselves to rank-based analysis. Second, even when only one rotational DoF exists, the associated translation component of the RBT is typically non-zero. Unlike translations, rotations belong to the special orthogonal group $SO(3)$, which is a curved manifold embedded in $\mathbb{R}^{3\times 3}$, not a linear space, so PCA cannot be applied. However, if the rotation is represented in axis-angle form, $r = \theta\omega$, where θ is the rotation angle and ω is the rotation axis, PCA can be applied.

Once the relative rotation matrices ${}^{l}R_{mk}$ are converted to axis-angle vectors ${}^{l}r_{mk}$, we construct the $3\times T$ matrix $\mathcal{R}_{lm} = [{}^{l}r_{m1}\ {}^{l}r_{m2}\ \ldots\ {}^{l}r_{mT}]$. If body m rotates relative to body l about a fixed axis (as in a revolute joint), then all ${}^{l}r_{mk}$ vectors lie along the same line in $\mathbb{R}^3$, implying $\mathrm{rank}(\mathcal{R}_{lm}) = 1$. This can again be verified using SVD.

However, even when only a single rotational DoF exists and no translational DoFs are present, the estimated RBTs may still exhibit non-zero translations ${}^{l}t_{mk}$. This occurs because the body does not rotate about the centroid of its keypoints, but about a separate pivot point. To address this, we estimate the effective center of rotation and test whether it remains consistent over time.

The effective center of rotation t'_{lk} satisfies the equation $t_{lk} = (R_{lk} - I_3)t'_{lk}$ when there is some rotation, that is $R_{lk} \neq I_3$. Therefore, we can estimate the center of rotation t'_{lk} by solving the equation $t_{lk} = At'_{lk}$, $A = R_{lk} - I_3$ for t'_{lk}. Since $\mathrm{rank}(A) = 2$ (as R_{lk}, being an odd-dimensional rotation matrix, always has one eigenvalue equal to 1), we solve using the Moore-Penrose pseudoinverse: $\hat{t}'_{lk} = A^+ t_{lk}$. Any point of the form $\hat{t}'_{lk} + \lambda_k\omega_l$ (where ω_l is the rotation axis) is also a valid center of rotation. If ω_l is constant, all $\hat{t}'_{lk}$ lie on a line in $\mathbb{R}^3$. Therefore, this condition can be identified by examining the dimensionality of the estimated centers of rotation $\hat{t}'_{lk}$ over time. Since the line along which these estimates lie does not necessarily pass through the origin, we construct the data matrix $\hat{\mathcal{T}}'_{lm} = [{}^{l}\hat{t}'_{m1} - \tau \quad {}^{l}\hat{t}'_{m2} - \tau \quad \ldots \quad {}^{l}\hat{t}'_{mT} - \tau]$, where each column represents the direction of a rotation center estimate relative to a reference point $\tau = {}^{l}\hat{t}'_{ma}$, chosen arbitrarily for some a such that $1 \leq a \leq T$. We then perform SVD on $\mathcal{T}_{lm}$, $\mathcal{R}_{lm}$, and $\hat{\mathcal{T}}'_{lm}$ for all $l = 0, \dots, \hat{n}$ and $m = 0, \dots, \hat{n}$, where frame 0 is the camera frame.

Let F be a symmetric matrix with entries f_{lm} representing the number of relative DoFs between each pair of bodies. The structure of the kinematic chain can be inferred from F. If $f_{0l} = 0$, then body l is static and part of the background. If $f_{0l_1} = 1$, then l_1 is the first link in the chain. If $f_{l_1 l_2} = 1$ for some l_2, then l_2 is the second link, and so on. This process continues until the full chain $[l_1, l_2, \dots, l_{\hat{n}}]$ is recovered. A similar approach can be extended to identify kinematic trees.

5 EMPIRICAL VERIFICATION

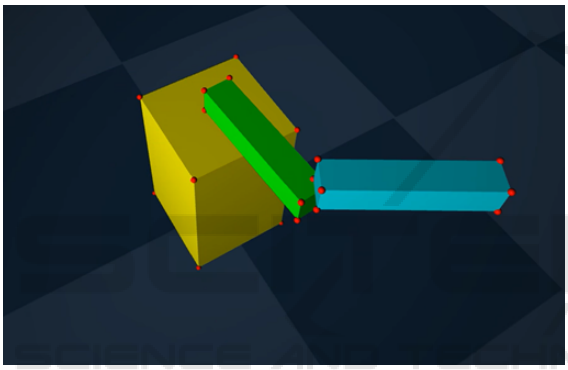

Figure 1: A 2-DoF arm on a base with keypoints (red dots) simulated and tracked by MuJoCo over 100 steps.

We conducted an empirical verification of the proposed algorithm using a simulated robot arm with two revolute joints in the MuJoCo physics engine (Todorov et al., 2012) (see Fig. 1). To isolate the algorithm's performance from that of a feature tracking system, we directly extracted the 3D positions of 23 keypoints (marked as red dots) from the simulator. These keypoints were placed at the vertices of three rigid bodies in the scene – the base and two links. MuJoCo's RGB-D rendering capabilities were used to determine the visibility of each keypoint from the camera's perspective at each time step. The robot arm was actuated by applying constant torques to its joints over a sequence of $T = 100$ control steps at a rate of 30 Hz, causing the links to complete approximately one full rotation each. The resulting keypoint visibility matrix over time is shown in Fig. 2.

The rigid body identification phase involved analyzing the pairwise distances between all keypoint pairs. Figure 3 shows the standard deviations of these

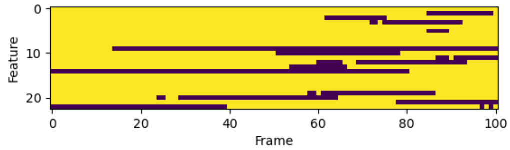

Figure 2: Visibility matrix of keypoints over time (yellow if visible, dark brown if not).

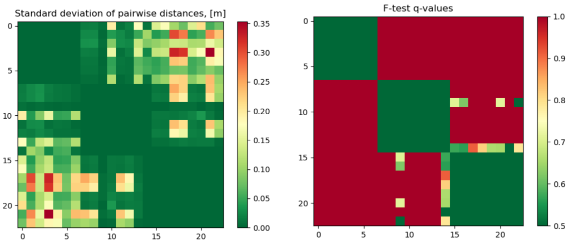

Figure 3: Standard deviations of all pairwise distances (left) and their associated q-values (right).

distances, along with the corresponding q-values obtained from the F-test, after introducing measurement noise of 1 mm. The results indicate that the F-test provides more uniform, robust, and discriminative dissimilarity metrics compared to raw distance variance.

Dendrograms resulting from agglomerative clustering with complete linkage using the two dissimilarity metrics are shown in Figs. 4 and 5, respectively. Both metrics do allow the successful identification of three well separated clusters that match exactly the ground truth (the point indices at the horizontal axis), but this process is much easier with the q-values than with standard deviations. To produce flat clustering from the dendrogram, the latter needs to be cut at a desired dissimilarity threshold. If standard deviations are used, any threshold between slightly above σ_{noise} and 0.19 would produce the correct number of clusters (three). However, any threshold higher than that would result in two clusters only, merging the points for the two links. Thus, the correct threshold depends on the range of motion of the rigid bodies in the scene, making it difficult to determine entirely from collected data. In contrast, the q-values lie in a standard range, where points that satisfy the RBA end up having dissimilarity around 0.5, and those that do not, end up with dissimilarity very close to 1. This makes it very easy to use a standard, domain-independent threshold for producing flat clustering, for example 0.8. Note also that the two points from the first link that are also somewhat close to some of the points in the second link do not present any problem for the clustering algorithm, as they are still closer to the points in the first link, so they are grouped with them, and later the high dissimilarity of those other points to the points in the second link prevent an incorrect merger of the two clusters.

Table 1: Non-zero singular values of transform matrices between pairs of body frames. F_0 is the world frame and frames F_l, $l = 1,2,3$ are the frames attached to each of the three discovered rigid bodies. Each table entry shows singular values for the translational, rotational, and center-of-rotation estimates for the relative RBTs. Colored cells indicate a single DoF.

	F_1			F_2			F_3		
F_0	-	-	-	2.9, 1.5	18.9	1.2	4.4, 3.1	19.6	10.2, 2.0, 0.4
F_1				2.9, 1.5	18.9	1.3	3.2, 2.8	19.6	10.1, 2.0, 0.4
F_2							6.0, 0.4	8.9	1.1

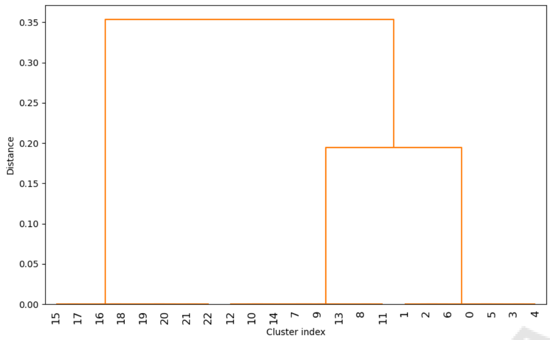

Figure 4: Dendrogram using standard deviations.

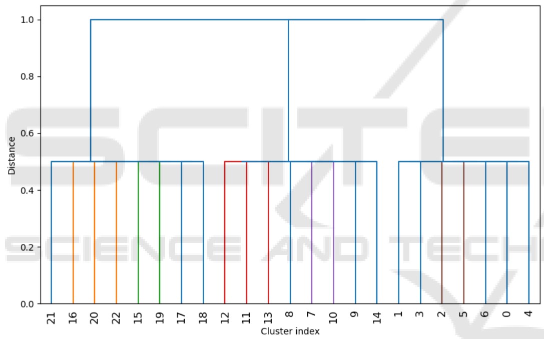

Figure 5: Dendrogram using q-values.

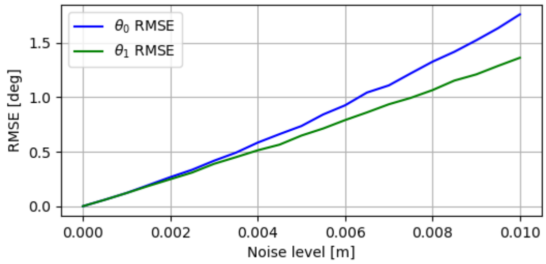

Figure 6: RMSE of joint angle estimates.

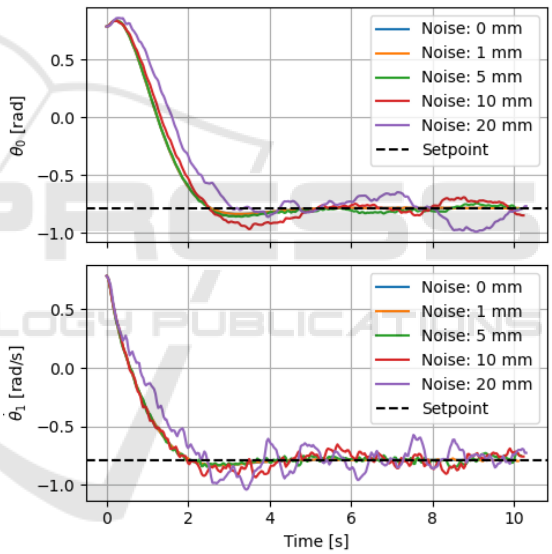

Figure 7: Joint angles under PBVS using OBIK with various levels of noise.

In the configuration construction phase, the ranks of the matrices $\mathcal{T}_{lm}$ and $\mathcal{R}_{lm}$ were computed for $l = 0,1,2$ and $m = l+1,\ldots,3$, with the results summarized in Table 1. Since no translational or rotational degrees of freedom were detected between bodies 1 (the base) and 0 (the world frame), it can be concluded that the base remains stationary. A single rotational DoF was identified between bodies 1 and 2, indicating that body 2 (the first link) rotates relative to the base. The presence of two translational DoFs between body 2 and both the world and base frames is explained by the fact that body 2 rotates around the end of the first link, tracing an arc in a plane – kinematically equivalent to 2D translation. This confirms that the second link is not directly connected to either the world or the base via a joint. However, when analyzing the DoFs of body 2 relative to the frame of body 1, the detection of a single rotational DoF confirms the presence of a revolute joint between the first and second links.

Because digital cameras have finite resolution, a key question concerning the use of the proposed method in practice is how quantization and other measurement errors of the 3D positions of keypoints would affect the accuracy of estimating the joint angles of the mechanism. To investigate this empirically, zero-mean Gaussian noise with various standard deviations was added to the keypoints' 3D positions and the resulting estimates of joint angles were compared with the true values reported by the simulator. The root mean-squared error (RMSE) of the estimates is shown in Fig. 6, averaged over 100 random-number seeds. The error is less than 0.02^o with noisefree keypoint position measurements, and grows only slightly faster than linearly with the level of keypoint noise. For measurement error of 1 mm, joint angle estimation error on the order of 0.21^o can be expected. This suggests that the proposed method, although certainly not as accurate as dedicated angular encoders on the joints, could be used for configuration estimation for the purposes of controlling the mechanism.

Ultimately, the most important question is how measurement noise in the 3D positions of keypoints affects the performance of a PBVS controller using the proposed OBIK scheme for configuration estimation. To investigate this, a proportional-derivative (PD) controller was applied to the 2-DOF mechanism with the objective of moving it from initial configuration $(\pi/4, \pi/4)$ to goal configuration $(-\pi/4, -\pi/4)$. The PD controllers for the two joints were independent, with proportional gains $K_{p1} = K_{p2} = 1$ and derivative gains $K_{d1} = 1$ and $K_{d2} = 0.5$ for links 1 and 2, respectively. The keypoint positions corresponding to the goal configuration were used as reference for the OBIK method, meaning that the PBVS PD controller was essentially trying to bring the estimated joint configuration to the origin, subject to measurement noise in the keypoint positions at each control step. Joint angle trajectories for the true angles of both joints, as recorded by the physics engine, are shown in Fig. 7 for several levels of measurement noise. The controller reaches the setpoint reliably and smoothly even for significant noise, as high as 20 mm. For noise on the order of 5 mm, which is already more than what is typical of modern RGB-D cameras, even in the depth dimension, the joint trajectories are virtually indistinguishable from those corresponding to when no measurement noise is present.

6 CONCLUSION AND FUTURE WORK

We introduced a method for learning compact representations of the configuration of articulated mechanisms from sequences of keypoint positions tracked in camera images. The approach relies on analyzing temporal variations in pairwise distances between keypoints to statistically determine which ones satisfy the RBA, thereby identifying groups that belong to the same rigid body. By examining the rank of matrices that capture the translational and rotational components of estimated poses over time, the algorithm infers the kinematic chain and the types of joints in it. The constructed configuration vector is as compact as that of the actual joint positions, effectively functioning as a joint observer without requiring prior knowledge of the mechanism's kinematics or appearance.

In future work, we aim to apply this observer to real-time monitoring and control of robotic systems and other articulated mechanisms. We also plan to investigate its robustness to noise and keypoint tracking errors, as in a real environment, changes in illumination and color, as well as lack of texture can lead to false matches and imprecise measurements.

REFERENCES

Chaumette, F., Hutchinson, S., and Corke, P. (2016). Visual servoing. *Handbook of Robotics*, pages 841–866.

Finn, C., Tan, X. Y., Duan, Y., Darrell, T., Levine, S., and Abbeel, P. (2016). Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519. IEEE.

Franklin, G. F., Powell, J. D., Emami-Naeini, A., and Powell, J. D. (2015). *Feedback control of dynamic systems*. Prentice hall Upper Saddle River.

Jonschkowski, R. and Brock, O. (2015). Learning state representations with robotic priors. *Autonomous Robots*, 39:407–428.

Kipf, T., Elsayed, G. F., Mahendran, A., Stone, A., Sabour, S., Heigold, G., Jonschkowski, R., Dosovitskiy, A., and Greff, K. (2021). Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*.

Lesort, T., Díaz-Rodríguez, N., Goudou, J.-F., and Filliat, D. (2018). State representation learning for control: An overview. *Neural Networks*, 108:379–392.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17:1–40.

Mojallizadeh, M. R., Brogliato, B., and Prieur, C. (2023). Modeling and control of overhead cranes: A tutorial overview and perspectives. *Annual Reviews in Control*, 56:100877.

Murtagh, F. and Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining*, 2(1):86–97.

Nelles, O. (2020). *Nonlinear dynamic system identification*. Springer.

Siciliano, B. and Khatib, O. (2016). *Springer handbook of robotics*. Springer International Publishing.

Szeliski, R. (2022). *Computer vision: algorithms and applications*. Springer Nature.

Todorov, E., Erez, T., and Tassa, Y. (2012). MuJoCo: A physics engine for model-based control. In *International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE.

Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380.

Wahlström, N., Schön, T. B., and Deisenroth, M. P. (2015). Learning deep dynamical models from image pixels. *IFAC-PapersOnLine*, 48(28):1059–1064.