Disentangled Object-Centric Configuration Representation Learning for Articulated Robot Arms

Daniel Nikovski[†]

Abstract— **The paper proposes a method for learning compact representations of the configuration (joint positions) of articulated mechanisms consisting of interconnected rigid bodies, from collected sequences of keypoint positions observed and tracked in camera images. The method analyzes the variations in pairwise distances between keypoints over time to deduce which of the keypoints must belong to the same rigid body and then computes the relative pose of all rigid bodies with respect to a reference image representing an initial or target configuration of the mechanism. By analyzing the rank of data matrices representing the translational and rotational components of the relative poses between the rigid bodies over time, the algorithm infers the order of the kinematic chain of the mechanism and the type of joints used in it, allowing the construction of a configuration vector as compact as the true joint positions of the mechanism.**

Index Terms— **Learning control, visual servoing, robotics**

I. Introduction

Monitoring and control of mechanisms requires an accurate representation of their current state that describes the position and velocity of all moving parts of the mechanism. For articulated mechanisms, such as robot arms, gantry cranes, etc. that consist of a number of rigid bodies connected by joints allowing relative motion between the bodies, the state is commonly expressed as a vector of the mechanism's configuration (joint positions or angles) and velocity (the joints' linear or angular velocities). Many articulated mechanisms are equipped with joint encoders that measure as accurately as possible the joints' positions and angles, but other mechanisms do not have such encoders, for cost or technical reasons, so direct measurements are not possible. Furthermore, even when encoders are available, they usually measure the angles of the motors actuating the joints, and intermediate mechanical components, such as belts and gearboxes, can introduce discrepancies between the joint and motor positions due to slip or gear backlash. Finally, other rigid bodies such as product parts manipulated by a robot arm could not possibly have encoders installed on them, but their configuration still needs to be known in order to execute a manipulation task. It is thus desirable to design alternative methods for state estimation that do not rely on encoders, but measure directly the configuration of a mechanism of interest.

A very appealing sensor modality for such configuration measurements is computer vision, due to the steadily decreasing costs of high-performance cameras and their increasing resolution and frame rates, allowing high-speed monitoring and control. The field of visual servocontrol (VS) [1] is concerned with the general problem of using camera images for servocontrol of mechanisms. Two major types of VS exist: image-based VS (IBVS) and position-based VS (PBVS). In IBVS, the control policy is conditioned directly on elements of the camera image, and in PBVS, the image is used to estimate explicitly the position (configuration) of the mechanism from the image, and execute a control law defined in terms of this configuration.

A major advantage of PBVS is that once the true state (joint positions and velocities) has been estimated, joint-space controllers of the mechanism can be used directly. For example, computed-torque controllers can use knowledge of the inverse dynamics of an articulated arm, if available, to provide advanced non-linear control that far outperforms linear controllers. However, a significant drawback of PBVS is that a state observer must be designed, which can be very laborious, difficult, and expensive.

In contrast, IBVS methods do not perform explicit state estimation, thus eliminating the need for state observers, and work directly with information extracted from the images. This information is typically in the form of the 2D image coordinates of various features of the image, such as corners, object centroids, etc. However, these methods usually make the assumption that all features of interest can be reliably tracked in every single image. This assumption is sometimes justified, for example when the camera is attached to the end tool of a robot arm (eye-in-hand setting) looking at a scene while the arm is approaching the goal position. In contrast, this assumption is typically not justified when the camera is observing the robot from a fixed position (eye-to-hand setting) and the features belong to the robot's links, as the links often occlude each other due to their close proximity and connectedness. For this reason, most IBVS method would fail in the eye-to-hand setting due to unreliable feature tracking.

A notable recent advance in IBVS is the emergence of methods for learning end-to-end visuo-motor policies that condition the control policy on the entire image, relying on a deep neural network (DNN) to extract the features and internal representations of the state of the mechanism that are necessary for its control. One example of such a method is the highly influential Guided Policy Search (GPS) algorithm [2]. However, the GPS algorithm assumes knowledge of the true internal state of system in order to compute a control law conditioned on this low-dimensional internal state using optimal control methods such as the iLQR algorithm [3], and after that trains a DNN to map high-dimensional images to the control outputs computed by the control law. Such knowledge of the true internal state of the mechanism is not available in the problem setting we are interested in, so the GPS algorithm and others of its class are not directly applicable.

Still, the basic idea of computing low-dimensional representations from images followed by control policies conditioned on these representations is applicable, and the field of state representation learning (SRL) has been concerned with devising efficient algorithms for this purpose [4]. SRL algorithms typically operate in a self-supervised learning setting, where the system first collects data during an exploration period under a suitable exploratory control policy whose purpose is to excite the system

[†] The author is with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA 02139 nikovski@merl.com

and make it visit the regions of its state space that are important for control, in line with the way system identification (SI) algorithms work. Some SRL algorithms construct state descriptors by employing an autoencoder convolutional DNN with a bottleneck layer, learning to reconstruct the image itself [5]. Other SRL algorithms learn both the state representation and the system dynamics jointly, similar to some SI algorithms, resulting in a nonlinear state-space model [6]. This kind of predictive models are widely used in model-based reinforcement learning (MBRL), where the much lower dimensionality of the constructed state space, compared to the very high dimensionality of the observation space (all pixels in an image), is a major factor in speeding up the computation of optimal sequential control policies by means of reinforcement learning algorithms [7].

Some notable successes with this approach in robotics include the swing-up of a cart-pole system, a difficult control benchmark problem even when the true state of the system is known [8], control of planar mobile robots [9], as well as object manipulation [5]. However, one difficulty associated with this approach is that although the learned representations are naturally distributed in the form of an activation pattern over all neurons in the bottleneck layer of a DNN, they are not necessarily disentangled (factored) over the individual moving bodies in the observed scene. This makes the learned representations impossible to compose and also affects adversely the sample complexity of learning, because all possible combinations of the states of all objects must be observed in the exploratory stage in order to learn a state representation that is valid everywhere in the state space of the system. This contrasts with the gradual way humans learn mental representations of the external world over their entire lifetimes, a few objects at a time along with their dynamics and affordances.

To address the lack of representation disentaglement associated with these earlier SRL approaches that were based on autoencoder DNNs, recent work has focused on object-centered SRL [10]. This approach recognizes the powerful inductive bias of learning a description of a scene consisting of separate descriptions of several individual objects that interact loosely with each other. This kind of object-centric inductive bias is very suitable to the problem we are considering, as articulated mechanisms indeed consist of multiple rigid bodies that are constrained to move in relation to each other only with one (or at most few) degrees of freedom (DoF).

The Slot Attention for Video (SAVi) method proposed in [10] uses a number of discrete slots, one per object, in combination with a transformer DNN, to learn disentangled object-centric representations. Similar to other SRL methods, the SAVi architecture takes as input sequences of images and relies on convolutional neural networks (CNNs) to extract features necessary for tracking and representing objects. Learning such features requires additional amounts of data and is known to be quite sensitive to proper initialization of the neural network, in particular its convolutional kernels. (The authors of [10] suggest the slots can be conditionally initialized based on cues such as the center of mass coordinates of objects, but this involves some type of pre-segmentation of the scene into objects.)

This observation raises the question of whether a neural network, with its associated long training time, high sample complexity, and less than perfect reliability is needed at all. We propose to use as input to the SRL algorithm not the raw pixels of camera images, but the spatial coordinates of a set of more-or-less reliably identifiable keypoint locations, and learn disentangled object-centric representations by analyzing the relative motion of these keypoints over time, as measured in a data set. This approach is similar to the one usually taken in the field of Structure from Motion (SfM) estimation and can leverage results from that area. The method identifies which keypoints must belong to the same rigid body using statistical tests, estimates the poses of the objects in the camera frame from the associated keypoints' coordinates in the form of rigid body transforms (RBT), and then identifies the order of the kinematic chain of the mechanism based on singular-value decomposition (SVD) of the relative RBTs between all pairs of identified bodies. This results in a compact description of the configuration of the articulated mechanism equivalent to the vector of joint positions or angles of the mechanism.

Section II describes the learning problem we are addressing and Section III explains the proposed algorithm for object-centric configuration learning. Section IV illustrates the algorithm with an example application to an articulated robot arm, and Section V proposes directions for future work and concludes the paper.

II. Problem Statement

We are interested in the problem of constructing a compact representation of the configuration of an articulated mechanism consisting of an unknown number of rigid bodies connected in a kinematic chain with single-DoF joints, from a sequence of camera images. The size and appearance of the rigid bodies defining the links of the mechanism are unknown, and neither is their order in the kinematic chain of the mechanism. The joints could be either prismatic or revolute. Under these conditions, it is not not possible to recover the true state of the mechanism in order to apply PBVS directly. However, if a configuration representation that is equivalent (has one-to-one mapping) to the true configuration of the mechanism can be constructed and estimated from camera data, PBVS methods can be applied to the control of the mechanism.

We assume that a set $\mathcal{P}$ of N distinctive spatial (3D) keypoints has been tracked over T instances in time producing measurements $p_{ik} = [x_{ik}, y_{ik}, z_{ik}]^T$, $i = 1, \ldots, N$, $k = 1, \ldots, T$. The matrix of measurements at time k is denoted by $P_k = [p_{ik}]^T \in \mathbb{R}^{N \times 3}$, $i = 1, \ldots, N$. One way to estimate the positions of these points is to use an RGBD (depth) camera to collect a sequence of T RGBD frames under a persistent excitation control policy for the target mechanism, and compute distinctive features, such as corners, in the RGB images [11]. Then, the positions p_{ik} of the features of interest can be computed in the camera's frame of reference by using the depth component of the RGBD camera's image and the intrinsic parameters of the camera. Note that we do not assume that all points are visible at all times; instead, for some time instances, some of the points might be occluded and thus their positions unknown. In addition, the measurement of the positions of those points which are visible is subject to measurement noise, due to the finite size of the camera's pixels and finite depth resolution, as well as possible mechanical disturbances. Furthermore, we assume that the number of points N is much larger than the number n of rigid bodies in the mechanism, as it is essential to have at least 3, and hopefully more, visible points per body in order to compute reliably its pose in the camera frame.

Given the (possibly sparse) data tensor $P = [P_k] \in \mathbb{R}^{N \times T \times 3}$, $k = 1, \ldots, T$ of position data, we want to determine how many rigid bodies exist in the scene, which of the N points belongs to which body, what are the poses of all bodies with respect to a reference pose at all times, which body is connected to which via a single joint, and what positions these joints have at all instants in time, relative to a reference joint position. The reference poses of the bodies could correspond to those the bodies have assumed at a specified moment in time in the sequence, for example the initial time instant, corresponding to the first camera image. For definiteness, the reference positions of the joints can be assumed to all have a value of zero at that time.

III. Configuration Learning Algorithm

A. Association of Points to Rigid Bodies

The first step of the algorithm is to determine how many moving bodies exist in the scene (including the stationary background) and to associate each point to exactly one body. The problem, as defined above, is an instance of the type of problems addressed in the field of SRL, but it also bears strong similarities to problems addressed in the field of Structure from Motion (SfM) estimation [11]. Typically, SfM algorithms work on 2D data, using only the projections of the tracked points onto the image plane, and aim to reconstruct the 3D coordinates and the relative motion between the moving objects and the camera. In our case, we do assume that we have access to the 3D coordinates, thus simplifying the problem.

We can use one of the major principles of SfM estimation formulated as the Rigid Body Assumption (RBA) by Ullman [12]: "any set of elements undergoing a two dimensional transformation which has a unique interpretation as a rigid body moving in space should be interpreted as such a body in motion". We can adopt the RBA as the main principle of determining how many bodies are moving in the scene by testing whether a pair of points maintains the same distance between each other over time, and cluster the set of N points into subsets that satisfy the RBA between each pair of its members.

In order to apply this approach, we need a measure of dissimilarity between each pair of points, expressing the degree to which the Euclidean distance between these two points violates the RBA. One possible dissimilarity measure is the variance of the distance D_{ij} between the two points p_i and p_j. This distance, under the assumption that measurement noise exists, is a random variable whose mean $\bar{d}_{ij}$ and sample variance s_{ij}^2 can be computed for the cases when both points are observable, as indicated by the indicator variables o_{ik} that equal 1 if point i is observable at time k and 0 if not:

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

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

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

In order to estimate the variances s_{ij}^2, the distance between the two points must be measured at least twice, so if the condition $\forall j, T_{ij} \geq 2$ is not satisfied for a point i, it is excluded from the set of points.

The sample variance s_{ij}^2 is one possible dissimilarity measure, but its scaling might not be very suitable for use by clustering algorithms. When two points do belong to the same moving object, the variance of the measured distance between them would be on the order of the measurement noise of the camera. When they do not belong to the same object, the variance would vary from slightly above this noise value (for cases when the two points do not move much to begin with), to much more than the noise. This creates a possibility for confusion by clustering algorithms, as they generally seek to minimize the total dissimilarity inside clusters and might erroneously consider points whose distance's variance is slightly above noise to belong to the same object.

A better dissimilarity measure might be derived if we cast the problem as one of statistical testing. Indeed, the objective here is to test the statistical hypothesis that the distance between two points (a random variable) is not constant, i.e., its variance is significantly higher than can be explained by the variance of the measurement noise alone. A suitable test for difference in variances is the F-test. To perform it, we need to know the variance of the measurement noise σ_{noise}^2. One way to estimate is from the specifications of the camera (viewing angle and resolution of RGB and depth images). Another way is to measure it empirically by tracking a set of points known to belong to the same object over multiple frames while the object is changing its pose, and computing the variance of the distances between visible points. It should be noted that the measurement noise will vary somewhat depending on the coordinates of the point, as the angle subtended by a single pixel changes depending on how far the pixel is from the optical axis of the camera, but for the purposes of statistical testing, a constant aggregate estimate of σ_{noise}^2 is generally sufficient.

The F-test proceeds by formulating a null hypothesis H_0 that the variance $\sigma_{ij}^2 = \sigma_{noise}^2$ and an alternative hypothesis H_A that $\sigma_{ij}^2 > \sigma_{noise}^2$. Here, we are computing a one-sided F-test, as the true variance σ_{ij}^2 can only be equal to or larger than the true noise variance σ_{noise}^2 (the latter is the Cramér-Rao bound to the former), so there is no physical possibility it can be lower. (For the respective measured values, we can sometimes have $s_{ij}^2 < \sigma_{noise}^2$, but when this happens, it clearly means the distance is constant, and we do not need further tests.)

The F-statistic is then computed as $F_{ij} = s_{ij}^2 / \sigma_{noise}^2$. It follows an F-distribution with degrees of freedom $(T_{ij} - 1, T_{ij} - 1)$. The p-value associated with the measured variance s_{ij}^2 can be computed as $p = Pr(F > F_{ij}) = 1 - CDF(F_{ij})$, where $CDF(\cdot)$ is the cumulative distribution function of the F distribution. The p-value signifies the probability that the observed F-statistic could be that high if the null hypothesis H_0 (the points belong to the same object) were true. A low value (below a chosen threshold) rejects the null hypothesis and suggests that the alternative hypothesis H_A (in this case, the points do not belong to the same object) is true at the chosen confidence level. For our purposes, though, we do not need to choose a confidence level – instead, we can use the complement of the p-value $q_{ij} = 1 - p = CDF(F_{ij})$ as the dissimilarity measure between points i and j for the purposes of clustering the points into separate objects. This dissimilarity measure falls in the general range between around 0.5 (when the measured variance s_{ij}^2 is approximately equal to the noise variance, so $F_{ij} \approx 1$), up to 1 (when $F_{ij} >> 1$). This more uniform scale makes it much easier to determine which points belong to the same object by means of clustering.

The clustering itself can be performed by means of any algorithm that takes as input a matrix of dissimilarity measures, such as hierarchical clustering, DBSCAN, spectral clustering, etc. Let $\mathcal{P}_l$, $l = 1, \hat{n}_0$ be the subsets that the points in $\mathcal{P}$ have been clustered into, and $N_l = |\mathcal{P}_l|$ be the number of points in cluster l. If $N_l < 3$, the corresponding subset is discarded, as it cannot be used for estimating the pose of the object it is associated with. The number $\hat{n}$ of remaining clusters is an estimate of the unknown true number n of rigid bodies in the scene, including the static background.

B. Construction of a Configuration Descriptor

Once the points have been assigned to clusters of size at least 3, we can compute the relative poses of all identified bodies with respect to a reference pose implied by the measured position of the points at a given time. Let these positions be denoted by $p_{i0} = [x_{i0}, y_{i0}, z_{i0}]^T$, and let $P_0 = [p_{i0}]^T \in \mathbb{R}^{N \times 3}$, $i = 1, \ldots, N$. A convenient setting could be to choose the measurements for the first frame in the sequence, i.e. $P_0 = P_1$, but any other frame can be used, too.

Let now $\mathcal{P}_{l0} = [\mathrm{p}_{0j}]_{j=1}^{N_l} = [[x_{0j}, y_{0j}, z_{0j}]^T]_{j=1}^{N_l}$ denote the matrix of reference measurements for cluster l, and $\mathcal{P}_{lk} = [\mathrm{p}_{ik}]_{i=1}^{n_l} = [[x_{ik}, y_{ik}, z_{ik}]^T]_{i=1}^{n_l}$ denote the matrix of measurements for the same points at time instance k, stacked horizontally with one column per point. If $N_l \geq 3$ and the points are in general position (not coplanar), we can estimate the RBT that maps the points in $\mathcal{P}_{l0}$ to those in $\mathcal{P}_{lk}$ optimally in least-square error sense: $\mathrm{p}_{ik} \approx R_{lk}\mathrm{p}_{i0} + t_{lk}$, by means of Procrustes superimposition using Kabsch-Umeyama's algorithm [13]. Here, R_{lk} is a 3×3 rotation matrix, and t_{lk} is a 3×1 translation vector. The method first computes the centroids c_{l0} and c_{lk} of both sets of points and translates the points so their centroids are at the origin: $\bar{\mathcal{P}}_{l0} = \mathcal{P}_{l0} - c_{l0}$ and $\bar{\mathcal{P}}_{lk} = \mathcal{P}_{lk} - c_{lk}$. It then computes the covariance matrix of the centered measurement matrices as $H = \bar{\mathcal{P}}_{l0}^T \bar{\mathcal{P}}_{lk}$, computes its SVD $H = USV^T$, and recovers the optimal rotation matrix $R_{lk} = VU^T$. Finally, the translation vector is recovered as $t_{lk} = c_{lk} - R_{lk}c_{l0}$.

After repeating the Procrustes superimposition procedure for each cluster/body, we obtain a set of $\hat{n}$ RBTs that completely define the configuration of the mechanism. This set is thus a valid constructed configuration descriptor that is of much lower dimensionality than that of the original RGBD images or key-points it was extracted from, so it can be used for monitoring and control of the mechanism, in principle. However, it is far from minimal. Each of the RBTs represents the 6-DoF relative pose of a body with respect to its reference configuration, with 3 rotational and 3 translational DoF. Thus, the entire configuration descriptor will have $6\hat{n}$ elements. However, if the mechanism is an open kinematic chain (for example, a robot arm), its configuration is defined only by $\hat{n}$ joint positions, i.e., 6 times more compact.

It is possible to further analyze the obtained RBTs of the identified rigid bodies in order to determine the kinematic structure of the mechanism and devise a more compact configuration descriptor. To this end, we can recognize that when two bodies are next to each other in the kinematic chain and are connected by a single-DoF joint, their *relative* RBT will have only a single DoF, too (either translational or rotational). Recall that the RBTs computed so far are expressed relative to a reference pose implied by the set of points $\mathcal{P}_{l0}$, all expressed in the inertial camera frame. The relative RBT between two objects l and m at time instant k can be expressed as the pose ${}^{l}R_{mk}$ of object m in the coordinate frame attached to object l. Thus, the relative rotation (orientation) would satisfy $R_{mk} = R_{lk}{}^{l}R_{mk}$, where leading superscripts denote the frame the rotation is expressed in, and rotations without such superscripts are the previously computed rotations with respect to the reference orientation in the camera frame. Then, the relative rotation can be obtained as ${}^{l}R_{mk} = R_{lk}^T R_{mk}$. Similarly, the translation component (relative position) can be obtained as ${}^{l}t_{mk} = R_{lk}^T(t_{mk} - t_{lk})$.

Note that these relative positions and poses are momentary and apply to a specific time instant k. How many degrees of freedom exist between the two bodies can be inferred by analyzing what happens to the relative pose over time. Trivially, if the relative position or orientation remains constant (within a testing tolerance), there are zero translational or rotational DoF between the two bodies. For most pairs of bodies, though, this will not be the case, and their relative pose will undergo changes over time.

Inferring the number of DoF in the translational component can be done by analyzing the rank of the $3 \times T$ matrix $\mathcal{T}_{lm} = [{}^{l}t_{m1}\ {}^{l}t_{m2}\ \ldots\ {}^{l}t_{mT}]$. If body m is undergoing only translational movement with respect to body l, each relative translation ${}^{l}t_{mk}$ should be along the constant axis of the same prismatic joint and thus should be a scalar multiple of all other translations. That is, all translations should lie on the same line (one-dimensional subspace) in $\mathbb{R}^3$, equivalent to rank$(\mathcal{T}_{lm}) = 1$. This condition can be detected easily by performing SVD on $\mathcal{T}_{lm}$.

Note that the condition rank$(\mathcal{T}_{mk}) = 1$ will also be true if the relative positions are constant (zero translational DoF between the two bodies), but not zero. For this reason, it is important to detect this condition beforehand, to avoid confusion with the case when an actual translation DoF does exist.

Discovering a single rotational DoF is more complicated, for two reasons. The first is that the representation of a relative orientation as a rotation matrix R does not lend itself naturally to a data matrix whose rank can reveal the existing DoF, and the second is that even when there is only a single rotational DoF, the translational component of the identifed RBT will generally not be zero. To illustrate the first reason, if we unfold the 9 entries of R and stack them in a matrix, this will not result in any special structure of the matrix when the rotations are the result of a single DoF. This is due to the fact that rotations, unlike translations, are not members of the 3D Euclidean space $\mathbb{R}^3$, but of the special orthogonal group $SO(3)$. It does not have the topology of an Euclidean space, but is a curved manifold embedded in $\mathbb{R}^{3\times 3}$, so linear principal component analysis on it is not informative.

However, there is still a way to recover low-dimensional structure in the set of estimated relative orientations if we represent them differently. What has Euclidean topology is the tangent space of the rotation R, expressed as its logarithm map: $\log(R) = \theta\widehat{\omega} \in \mathfrak{so}(3)$, where θ is the angle of rotation and $\widehat{\omega}$ is a skew-symmetric matrix whose elements are the coordinates of the unit vector ω corresponding to the axis of rotation. The Lie algebra $\mathfrak{so}(3)$ is a vector space and as long as the axis of rotation remains the same, its logarithm moves along a straight line. We can use this property to perform PCA in that space. Although, instead of using the skew-symmetric matrix $\widehat{\omega}$, we will use directly the rotation axis ω, representing the rotation as the product $r = \theta\omega$. This is also known as the axis-angle representation of a rotation.

Once the rotation matrices are transformed in this representation, analysis proceeds analogously to the translation case. The axis-angle representations ${}^{l}r_{mk}$ of the relative rotations ${}^{l}R_{mk}$ are placed in the $3 \times T$ matrix $\mathcal{R}_{lm} = [{}^{l}r_{m1}\ {}^{l}r_{m2}\ \ldots\ {}^{l}r_{mT}]$. If body m is undergoing only rotational movement with respect to body l, each relative rotation ${}^{l}r_{mk}$ should be along the constant axis of the same revolute joint and thus should be a scalar multiple of all other rotations. That is, all rotations should lie on the same line (one-dimensional subspace) in $\mathbb{R}^3$, equivalent to $\text{rank}(\mathcal{R}_{lm}) = 1$. This condition can again be detected easily by performing SVD on $\mathcal{R}_{lm}$.

However, as noted above, even when there is only a single rotational DoF between a pair of bodies and no translational DoFs, the computed sequences of RBTs will generally have a non-zero translational part ${}^{l}t_{mk}$, too. This is due to the fact that the body the keypoints belong to does not rotate around the centroid of these keypoints, but around some other pivot point. This creates the possibility of identifying one or more false translational DoFs even when there are none.

Fortunately, this condition can be ruled out by computing the effective center of rotation for all time steps and testing whether it remains the same, as it should, if there is only one rotational DoF with a constant center and axis of rotation. Recall that we estimated the rotation R_{lk} and translation t_{lk} that map any reference point p_{i0}, $i = 1, \ldots, N_l$ that we determined belongs to body l to its position p_{ik} at time k, both measured in a common frame (for example, the world frame), such that $\mathrm{p}_{ik} \approx R_{lk}\mathrm{p}_{i0} + t_{lk}$. Also, for any point ${}^{l}\mathrm{p}_i$ expressed in frame l, its position (constant over time in l) can be expressed in the world frame as

$$\mathrm{p}_{ik} = R'_{lk}\ {}^{l}\mathrm{p}_i + t'_{lk}, \tag{4}$$

where R'_{lk} is the rotation matrix expressing the rotation of body l in the world frame and t'_{lk} is the center of rotation of this joint. As we do not have access to measurements of the joint angle, we can set its value for the reference time to be such that $R'_{l0} = I_3$. Then, for that moment, we will have

$$\mathrm{p}_{i0} = I_3\ {}^{l}\mathrm{p}_i + t'_{l0}. \tag{5}$$

Suppose now that this motion was due to pure rotation of the frame attached to body l. This means that the rotation matrix R'_{lk} changes over time, but the center of rotation remains the same: $t'_{l0} = t'_{lk}$, $k = 1, \ldots, T$. Using this and (5), we express the unknown position of point i in the frame of body l as ${}^{l}\mathrm{p}_i = \mathrm{p}_{i0} - t'_{lk}$ and substitute in (4), yielding $\mathrm{p}_{ik} = R'_{lk}\mathrm{p}_{i0} + (R'_{lk} - I_3)t'_{lk}$. By comparing the terms in the approximate equation $\mathrm{p}_{ik} \approx R_{lk}\mathrm{p}_{i0} + t_{lk}$ that relates the same measurements p_{ik} and p_{i0} in the reference frame, we can identify that $R'_{lk} \approx R_{lk}$ and $t_{lk} \approx (R'_{lk} - I_3)t'_{lk}$. That means that the rotational matrix R_{lk} we estimated by the Kabsch-Umeyama method is indeed the best possible estimate of R'_{lk}. However, t_{lk} and t'_{lk} are not equal, but only indirectly related. We can still estimate the center of rotation by solving the equation $t_{lk} = At'_{lk}$, $A = R'_{lk} - I_3$ for t'_{lk}, when $R'_{lk} \neq I_3$.

However, even when $R'_{lk} \neq I_3$, we have $\text{rank}(A) = 2$, because a rotation matrix in odd dimensions always has one eigenvalue equal to 1, causing A to have at least one zero eigenvalue. We can still solve the equation as $\hat{t}'_{lk} = A^{+}t_{lk}$, using the Moore-Penrose pseudoinverse A^{+}. Note that any rotation center of the form $\hat{t}'_{lk} + \lambda_k\omega_l$, where ω_l is the axis of rotation, is also a solution to the equation, reflecting the physical reality that any point on the axis of rotation is a valid rotation center. However, as long as the axis of rotation ω_l remains constant in the reference frame (as it will for a single rotational DoF for body l), all estimates $\hat{t}'_{lk}$ will lie on the same line in 3D vector space. Thus, this condition can be detected by analyzing the dimensionality of estimates $\hat{t}'_{lk}$ over time. Recognizing that this line does not necessarily go through the origin, we form the data matrix $\hat{\mathcal{T}}'_{lm} = [{}^{l}\hat{t}'_{m1} - \tau\ \ {}^{l}\hat{t}'_{m2} - \tau\ \ \ldots\ \ {}^{l}\hat{t}'_{mT} - \tau]$ of the *directions* ${}^{l}\hat{t}'_{mk} - \tau$ of the rotation center estimates, expressed in reference frame l, with respect to one of these estimates $\tau = {}^{l}\hat{t}'_{ak}$ for some arbitrary a, $1 \leq a \leq T$.

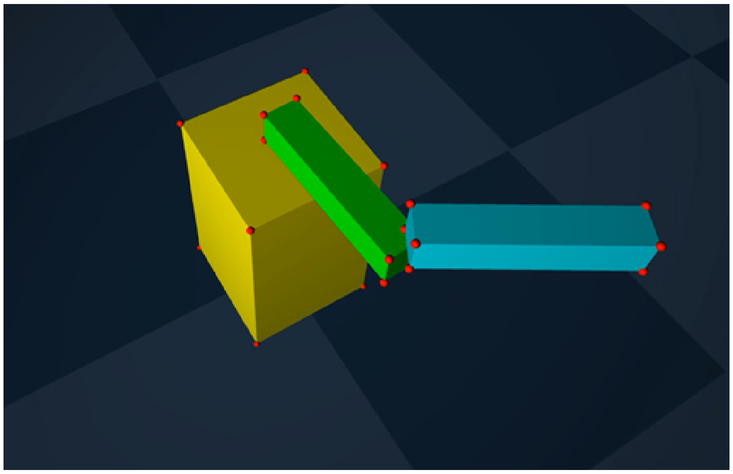

Fig. 1: A 2-DoF arm on a base with keypoints tracked by MuJoCo (shown with red dots), simulated for 100 steps.

The SVD of the matrices $\mathcal{T}_{lm}$, $\mathcal{R}_{lm}$, and $\hat{\mathcal{T}}'_{lm}$ is performed for all $l = 0, \ldots, \hat{n}$, $m = 0, \ldots, \hat{n}$, where the frame of reference 0 is the camera frame. Let the symmetric matrix F with entries f_{lm} contain the discovered number of relative DoFs between all pairs of bodies. We can discover the order of the kinematic chain of the mechanism by analyzing these entries, as follows. If $f_{0l} = 0$ for some l, then body l is stationary and belongs to the background. If $f_{0l_1} = 1$ for some l_1, then body l_1 is the first link in the kinematic chain. Then, if $f_{l_1 l_2} = 1$ for some l_2, then body l_2 is the second link in the kinematic chain. Analysis proceeds analogously until the entire kinematic chain is identified as the sequence $[l_1, l_2 \ldots, l_{\hat{n}}]$. A variant of this procedure can be applied to recover a kinematic tree, too.

IV. Empirical Evaluation

We performed an empirical evaluation of the proposed algorithm using a robot arm with two revolute joints simulated in the physics engine MuJoCo [14] (Fig. 1). In order to evaluate the performance of the algorithm independently of the performance of the tracker that would measure feature points and spatial locations, we retrieved directly from the simulator the locations of 23 keypoints (shown as red dots) attached to the corners of three rigid bodies in the scene (the base and two links), and used MuJoCo's RGBD rendering capabilities to determine which of them are visible by the camera at a given time. Constant torques

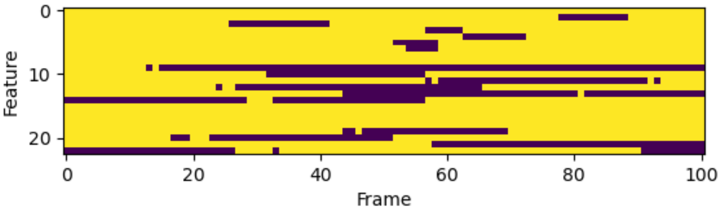

Fig. 2: Visibility matrix of feature keypoints over time (yellow if visible, dark brown if not).

were applied to the arm's joints over a sequence of $T = 100$ control steps at control rate of 30 Hz, making the links sweep approximately one full rotation. The resulting visibility matrix of the keypoints over time is shown in Fig. 2.

The rigid body identification step proceeded with analyzing the pairwise distances between all pairs of keypoints. Fig. 3 shows the standard deviations of the distances and the q-values from the F-test, after adding measurement noise of 1 mm. The results suggest that the F-test is a more reliable dissimilarity measure.

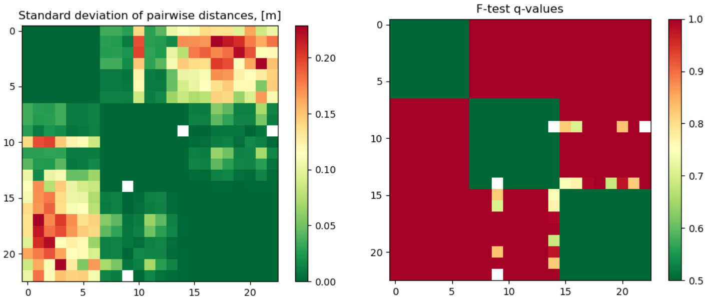

Fig. 3: Standard deviations of all pairwise distances (left) and their associated q-values (right).

During the configuration construction step, the ranks of the matrices $\mathcal{T}_{lm}$ and $\mathcal{R}_{lm}$ were computed for $l = 0, 1, 2$ and $m = l + 1, \ldots, 3$ and are shown in Table I. Because neither translational nor rotational DoFs were found between body 1 (the base) and the world frame (0), it can be concluded that the base is stationary. One rotational DoF was discovered between bodies 1 and 2, concluding that body 2 (the first link) rotates with respect to the base. The two translational DoF of the second link (body 2) with respect to the world and base frames is due to the fact that it rotates about the end of the first link, which sweeps an arc in the plane, equivalent to 2D translation. This confirms that the second link is not directly connected by a joint to either the world frame or the base. However, once the DoFs of the second link are computed from the RBT with respect to the frame of the first link, the single rotational DoF discovered identifies the existence of a revolute joint between the first and second link.

V. Conclusion and Future Work

A method was proposed for learning compact representations of the configuration of articulated mechanisms from collected sequences of keypoint positions observed and tracked in camera images. The method analyzes the variations in pairwise distances between keypoints over time to deduce via statistical testing which of the keypoints satisfy the RBA and must belong to the same rigid body. It then computes the relative poses of all rigid bodies with respect to a reference image and by analyzing the rank of data matrices representing the translational and rotational components of these relative poses over time, the algorithm infers the order of the kinematic chain of the mechanism and the type of joints used in it. This allows the construction of a configuration vector as compact as the true joint angles and positions, essentially creating a joint angle/position observer without any knowledge of the mechanism's kinematics or appearance.

	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

TABLE I: Non-zero singular values of transform matrices between pairs of body frames. Frame 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 pair shows singular values for the translational, rotational, and center-of-rotation estimates for the relative RBTs.

In future work, we plan to use this observer for monitoring and control of robots and other mechanisms. We plan to investigate how its performance will be affected by noise and tracking errors of keypoints. Although the algorithm was described in terms of using 3D keypoint positions, it might also be possible to leverage SfM algorithms to extend it to operate on planar images collected by one or more regular RGB cameras, too.

References

[1] F. Chaumette, S. Hutchinson, and P. Corke, "Visual servoing," *Handbook of Robotics, 2nd ed.*, pp. 841–866, 2016.

[2] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, pp. 1–40, 2016.

[3] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems," in *First International Conference on Informatics in Control, Automation and Robotics*, vol. 2. SciTePress, 2004, pp. 222–229.

[4] T. Lesort, N. Díaz-Rodríguez, J.-F. Goudou, and D. Filliat, "State representation learning for control: An overview," *Neural Networks*, vol. 108, pp. 379–392, 2018.

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

[6] N. Wahlström, T. B. Schön, and M. P. Deisenroth, "Learning deep dynamical models from image pixels," *IFAC-PapersOnLine*, vol. 48, no. 28, pp. 1059–1064, 2015.

[7] T. M. Moerland, J. Broekens, and C. M. Jonker, "Model-based reinforcement learning: A survey," *arXiv preprint arXiv:2006.16712*, 2020.

[8] J. Mattner, S. Lange, and M. Riedmiller, "Learn to swing up and balance a real pole based on raw visual input data," in *Neural Information Processing: 19th International Conference, ICONIP 2012, Doha, Qatar, November 12-15, 2012, Proceedings, Part V 19*. Springer, 2012, pp. 126–133.

[9] R. Jonschkowski and O. Brock, "Learning state representations with robotic priors," *Autonomous Robots*, vol. 39, pp. 407–428, 2015.

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

[11] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.

[12] S. Ullman, "The interpretation of structure from motion," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 203, no. 1153, pp. 405–426, 1979.

[13] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 04, pp. 376–380, 1991.

[14] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.