Articulation Estimation Using Depth Sensing

Suren Kumar
Mechanical and Aerospace Engineering
State University of New York at Buffalo
Buffalo, NY, USA
Email: surenkum@buffalo.edu

Vikas Dhiman
Electrical Engineering
University of Michigan
Ann Arbor, MI, USA
Email: dhiman@umich.edu

***Abstract*—**

- **Detect distinctly moving clustered points/voxels/objects in a scene.**
  - **Use Kinect fusion to create a static map.**
  - **Use some kind of noise threshold to detect object movement independent of camera movement.**
  - **Trigger algorithm (may be use RANSAC ?? etc.) that will segment out the object that just moved. The object should be spatially clustered and should be explained by the same rigid 3D motion.**
  - **Maintain a pairwise relative localization graph of the scene.**
- **Semantic reasoning in map update of these objects and their localization. Reason about Physical support and articulated linkage.**
- **Build a 3D reconstruction of these objects.**
- **Find similar unmapped static objects in the scene. May be use Jeff's detection and segmentation code.**
- **Try algorithm for long term mapping (a week) by using auto charging turtlebots in a living room and compare with existing algorithms.**

I. Introduction

Imagine a robot moving in a typical living room environment which encounters indoor objects such as doors, drawers and chairs etc. We posit that in order for the robot to understand, map or interact with such objects, the robot needs to be able to understand the articulation. Pyschophysical experiments on human motion understanding have demonstrated that human first distinguish between competing motion models (translation, rotation and expansion) and then estimate the motion conditioned on motion model [1].

II. Related Work

A. Historical Perspective

Using image motion to understand motion and structure in the scene is a historically well studied problem in computer vision. Ullman [2] proposed that in non-degenerate cases under orthographic projection, three pictures of four points can determine structure and motion. Tomasi and Kanade [3] formalized Ullmans's idea and proposed one of the influential method to compute camera motion and image structure by tracking features in the images. They proposed factorizing a matrix of feature tracks into motion and shape matrix by enforcing the rank constraint of the rigid body motion and metric constraints of a rotation matrix. Costeira and Kanade [4] extended the factorization idea to segment and recover shape along with motion of multiple moving bodies in the scene.

Yan and Pollefeys [5] further extended this rank and subspace idea to estimate kinematic chains from tracked features. There are certain fundamental limitations which is common to these approaches. First, the reliance on feature tracking methods such as KLT is not suitable for indoor environments which may not have much texture. Furthermore, this feature tracking requirement limits the application of such methods tremendously by i) Not being able to track new parts/objects that enter/exit the scene, ii) Not modelling the entire scene and as a result not exploring dependencies between neighbouring objects, iii) Restricting the ability to assimilate the motion of objects in SLAM like approaches that map the entire scene. Secondly, motion orthogonal to image plane is not modelled [5] as image projection is modelled as affine projection in the most general case.

With the discovery of cheap and commonplace hardware such as Kinect, there is a need to re-examine this structure from motion idea. First, since such hardware already provides depth for a feature point, one already has shape as estimated by traditional structure from motion. Also since depth is available, one can model the motion orthogonal to image plane. Textureless objects can be tracked by adding depth edges to the tracking mix.

[6] Build on existing work on articulation estimation. Add interactive perception where manipulation adds to perception and vice versa. [5] Problem: Analysis and reconstruction of dynamical scenes Method: Estimate the rigid motion subspaces. Extend the method to non-rigid parts by modeling it with linear combination of key shapes. Use motion segmentation (by SVD) to segment feature trajectories for each object. Use n neighbors to estimate local subspace of each point and cluster the subspaces to estimate the cluster of trajectories that form the same metric subspace.

[**?**] uses RGB for 3D articulation estimation. [7] Uses RGBD

TODO: Anguelov et al. - kinematic models of doors
TODO: Yan and Pollefeys : Assumes affine geometry. Only revolute joints.

TODO: [8] : 3D trajectories by manipulation of environment
TODO: [9]
TODO: [10]

III. Articulation Classification

We consider the problem of motion model identification from point correspondences of motion. In the current work, we consider revolute, prismatic and general motion. Consider motion of two points x_0, x_1 (represented in an inertial frame) on a rigid body at time t_0 and at some subsequent times t_1, t_2. The most general form of rigid body motion of a point can be represented using a rotation matrix R and an associated translation vector T

$$x_0^{t_1} = R_{t_0}^{t_1} x_0^{t_0} + T_{t_0}^{t_1} \tag{1}$$

where the superscript on the point denotes the time.

A. Prismatic

For points lying on a prismatic joint such as a drawer, rotation w.r.t inertial frame remains the same (R=I), resulting in $x_1^{t_1} - x_0^{t_1} = x_1^{t_0} - x_0^{t_0}$. This is essentially saying that the vector joining two points on a prismatic joint remains the same before and after the motion.

B. Revolute

For further distinction between revolute and general motion, we need information from more than one time step. For points lying on a body undergoing revolute motion such as a door, the points have same translation vector over time. Hence estimating the translation vector from two time steps $T_{t_0}^{t_1} = T_{t_1}^{t_2}$ is a sufficient condition to classify a joint as revolute joint.

C. Plane Constrained Motion

Plane constrained motion is useful for characterizing motion of objects like chair that can be translated on a plane and rotated about the normal to the plane. Let the plane be denoted by an point x_p lying on the plane and $\hat{n}$ being normal to that plane. Consider the case of a rigid body that has point $x_c^{t_0}$ in contact with the ground plane which after undergoing the motion moves to $x_c^{t_1}$. Since $x_c^{t_0}$ and $x_c^{t_1}$ both lie on the ground plane, we have

$$(x_c^{t_0} - x_0)^T \hat{n} = 0 \tag{2}$$

$$(x_c^{t_1} - x_0)^T \hat{n} = 0 \tag{3}$$

$$x_c^{t_1} = R_{t_0}^{t_1} x_c^{t_0} + T_{t_0}^{t_1} \tag{4}$$

By doing algebraic manipulation we get, $(R_{t_0}^{t_1} x_0 + T - x_0)^T \hat{n} = 0$

D. General Rigid Body Motion

For general motion such as a book that can be rotated and translated anywhere in the space both the rotation and translation matrix will be different.

IV. Scene Understanding

Analysis by method such as ours is essential in order to decompose the scene into types of motion that a robot can influence on the scene. For example: Understanding the way a drawer can be opened, fridge door can be opened, what can be moved around in the scene

Other important use cases of motion estimation

- Estimation a joint can induce prior over objects such as revolute joint can induce prior over refrigerator and door
- Visual object identification such as drawer can induce a prior over motion estimation
- Object tracking
- For grasping? – such as how to open a door?

V. Factorization Approach

In this section, we extend the factorization approach as described in [4] to 3-D track data available from a depth camera. Assuming for now that a single object moves relative to a static camera and we track features from frame to frame. Following the notation in the paper, lets represent a point on the object as $p_i^T = [X_i, Y_i, Z_i]^T$ in the camera frame, in the current frame f, the position of the point in homogeneous coordinates can be represented as

$$s_{fi}^C = \begin{bmatrix} p_{fi}^C \\ 1 \end{bmatrix} = \begin{bmatrix} R_f & t_f \\ 0_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} p_i \\ 1 \end{bmatrix} = \begin{bmatrix} R_f & t_f \\ 0_{1\times 3} & 1 \end{bmatrix} s_i \tag{5}$$

where R_f and t_f are the rotation and translation of the object from current frame w.r.t to the frame in which object points are initially represented. Assuming that we track N features over F frames, one can write

$$\begin{bmatrix} u_{11} & \dots & u_{1N} \\ . & & . \\ u_{F1} & \dots & u_{FN} \\ v_{11} & \dots & v_{1N} \\ . & & . \\ v_{F1} & \dots & v_{FN} \\ w_{11} & \dots & w_{1N} \\ . & & . \\ w_{F1} & \dots & w_{FN} \end{bmatrix} = \left[\begin{array}{c|c} i_1^T & t_{x_1} \\ . & . \\ i_F^T & t_{x_F} \\ j_1^T & t_{y_1} \\ . & . \\ j_F^T & t_{y_F} \\ k_1^T & t_{z_1} \\ . & . \\ k_F^T & t_{z_F} \end{array}\right] \begin{bmatrix} s_1 & . & . & . & s_N \end{bmatrix} \tag{6}$$

where (u_{fi}, v_{fi}, w_{fi}) is the location of feature point in current frame, vectors i_f^T, j_f^T, k_f^T are the rows of the rotation matrix R_f and $(t_{x_f}, t_{y_f}, t_{z_f})$ represent the components of the translation vector at time instant with frame f. Equation 6 can be represented in a accumulated form as

$$\mathbf{W} = \mathbf{MS} \tag{7}$$

where $\mathbf{W}$ represents the accumulation from trajectories of N points tracked over F frames, $\mathbf{M}$ contains all the information about the motion of the object present in the scene and $\mathbf{S}$ contains all the information about the shape of the object. Since rank of product of two matrices can not exceed the minimum of rank of individual matrics, It is clear that the maximum rank of W is 4. Computing singular value decomposition of $\mathbf{W}$, we get

$$\mathbf{W} = \mathbf{U}\Sigma\mathbf{V}^T \tag{8}$$

where $U \in R^{3F\times 4}$, and $V \in R^{N\times 4}$ are left and right real singular matrices and Σ is a 4×4 diagonal matrix of singular values. Although if $\mathbf{W}$ was full rank, we would have to consider N singular values but as the rank of W is 4, we only write out the components corresponding to first 4 singular values. Writing the factorization as product of two matrices,

$$\hat{\mathbf{M}} = \mathbf{U}\Sigma^{\frac{1}{2}}, \hat{\mathbf{S}} = \Sigma^{\frac{1}{2}}\mathbf{V}^T \tag{9}$$

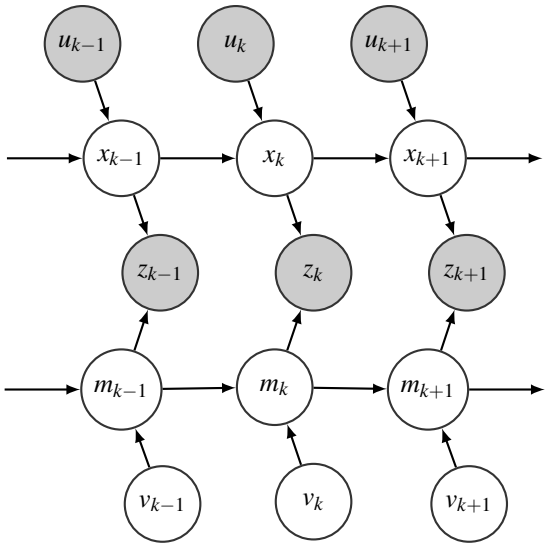

Fig. 1: Graphical Model of the general SLAM problem. The known nodes are darker than the unknown nodes.

A. Motion and Shape Estimation

The factorization as defined in Equation 9 is not unique as any invertible 4×4 matrix A will lead to an alternate solution $\mathbf{M}=\hat{\mathbf{M}}A$, $\mathbf{S}=A^{-1}\hat{\mathbf{S}}$

$$\begin{bmatrix} m_{00}^2 & 2m_{00}m_{01} & 2m_{00}m_{02} & m_{01}^2 & 2m_{01}m_{02} & m_{02}^2 \\ m_{10}m_{00} & m_{10}m_{01}+m_{11}m_{00} & m_{10}m_{02}+m_{12}m_{00} & m_{11}m_{01} & m_{11}m_{02}+m_{12}m_{01} & m_{12}m_{02} \\ m_{20}m_{00} & m_{20}m_{01}+m_{21}m_{00} & m_{20}m_{02}+m_{22}m_{00} & m_{21}m_{01} & m_{21}m_{02}+m_{22}m_{01} & m_{22}m_{02} \\ m_{00}m_{10} & m_{00}m_{11}+m_{01}m_{10} & m_{00}m_{12}+m_{02}m_{10} & m_{01}m_{11} & m_{01}m_{12}+m_{02}m_{11} & m_{02}m_{12} \\ m_{10}^2 & 2m_{10}m_{11} & 2m_{10}m_{12} & m_{11}^2 & 2m_{11}m_{12} & m_{12}^2 \\ m_{20}m_{10} & m_{20}m_{11}+m_{21}m_{10} & m_{20}m_{12}+m_{22}m_{10} & m_{21}m_{11} & m_{21}m_{12}+m_{22}m_{11} & m_{22}m_{12} \\ m_{00}m_{20} & m_{00}m_{21}+m_{01}m_{20} & m_{00}m_{22}+m_{02}m_{20} & m_{01}m_{21} & m_{01}m_{22}+m_{02}m_{21} & m_{02}m_{22} \\ m_{10}m_{20} & m_{10}m_{21}+m_{11}m_{20} & m_{10}m_{22}+m_{12}m_{20} & m_{11}m_{21} & m_{11}m_{22}+m_{12}m_{21} & m_{12}m_{22} \\ m_{20}^2 & 2m_{20}m_{21} & 2m_{20}m_{22} & m_{21}^2 & 2m_{21}m_{22} & m_{22}^2 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{01} \\ a_{02} \\ a_{11} \\ a_{12} \\ a_{22} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{10}$$

VI. SLAM for Dynamic World

Figure 1 shows the graphical model of the most general SLAM problem, where x_k, u_k, z_k, m_k, v_k represents the robot state, input to the robot, observation by robot, state of the world and action of various agents in the environment.

Basic SLAM algorithms *assume the map* $m_{k-1}\equiv m_k\equiv m$ *to be static* and model the combination of robot state and map x_k, m as the state of the estimation problem. The estimation problem only requires motion model $P(x_k|x_{k-1},u_k)$ and observation model $P(z_k|x_k,m)$. The observation model assumes the observations to be conditionally independent given the the map and the current vehicle state. The goal of the estimation process is to produce unbiased and consistent estimates (expectation of mean squared error should match filter-calculated covariance) [11].

For the current SLAM problem, the state consists of time-varying map, (unknown input to the world by various agents) and the robot state. Hence the full estimation problem can be posed as

$$P(x_k,m_k|\mathbf{Z}_{0:k},\mathbf{U}_{0:k},\mathbf{V}_{0:k},x_0,m_0) \tag{11}$$

Following the notation in the review paper on SLAM by Durrant-Whyte and Bailey [12], $\mathbf{Z}_{0:k}$, $\mathbf{U}_{0:k}$ and $\mathbf{V}_{0:k}$ represent the set of observations, robot control inputs and map control inputs from the start time to time step k. It is assumed that the map is markovian in nature which implies that the start state of the map m_0 has all the information needed to make future prediction if actions of various agents in the world $v_{k-1},...,v_{k+1}$ and its impact on the map is known.

A. Time update

The time update models the evolution of state according to the motion model. To write equation concisely, let $A=\{\mathbf{Z}_{0:k-1},\mathbf{U}_{0:k},\mathbf{V}_{0:k},x_0,m_0\}$

$$\begin{aligned} &P(x_k,m_k|A)= \\ &\int\int P(x_k,x_{k-1},m_k,m_{k-1}|A)dx_{k-1}dm_{k-1} \\ &\int\int P(x_k|x_{k-1},m_k,m_{k-1},A)P(x_{k-1},m_k,m_{k-1}|A)dx_{k-1}dm_{k-1} \\ &\int\int P(x_k|x_{k-1},u_k)P(x_{k-1},m_k,m_{k-1}|A)dx_{k-1}dm_{k-1} \\ &\int\int P(x_k|x_{k-1},u_k)P(m_k|x_{k-1},m_{k-1},A)P(x_{k-1},m_{k-1}|A)dx_{k-1}dm_{k-1} \\ &\int\int P(x_k|x_{k-1},u_k)P(m_k|m_{k-1},v_{k-1})P(x_{k-1},m_{k-1}|A)dx_{k-1}dm_{k-1} \end{aligned} \tag{12}$$

The independence relationship in derivation of time update in Equation 12 are due to the Bayesian networks in Figure 1 in which each node is independent of its non-descendants given the parents of that node. Given the structure of time update, we need two motion models, one for robot: $P(x_k|x_{k-1},u_k)$ and another one for the world $P(m_k|m_{k-1},v_{k-1})$. It can be clearly observed that $P(m_k|m_{k-1},v_{k-1})$ for a static map is dirac delta function and integrates out in Equation 12.

B. Measurement Update

Measurement update uses the bayes formula to update the state of the estimation problem given a new observation z_k at time step k. To write the equations concisely, let $B=\{\mathbf{Z}_{0:k},\mathbf{U}_{0:k},\mathbf{V}_{0:k},x_0,m_0\}$

$$\begin{aligned} P(x_k,m_k|B) &= \frac{P(z_k|x_k,m_k,A)P(x_k,m_k|A)}{P(z_k|A)} \\ &= \frac{P(z_k|x_k,m_k)P(x_k,m_k|A)}{P(z_k|A)} \end{aligned} \tag{13}$$

Equation 13 together with equation 12 defines the complete recursive form of the SLAM algorithm for a dynamic environment. Robot motion model and observation model $P(z_k|x_k,m_k)$ are well described in previous literature and hence we will exclude that from current discussion. The focus of current work is the representation of map motion model to extend the standard SLAM algorithm with its static world assumption to dynamic world.

VII. Dynamic World Representation

Real world is dynamic in nature with varying degree of motion such as parking lot which can be assumed to be temporary stationary compared to a road which is always in motion. Previous literature to handle dynamic environments

can be divided into two predominant approaches A) Detect moving objects and ignore them, B) Track moving objects as landmarks [13]. In the first approach, using the fact that the conventional SLAM map is highly redundant, the moving landmarks can be removed from the map building process [14]. In contrast, Wang et. al [15] explicitly track moving objects by adding them to the estimation state. However the work assumed that the sensor measurement can be decomposed into observation corresponding to moving and static landmarks which requires good estimate of moving and static landmarks to start with. Furthermore, it was assumed that the measurement of moving object carries no information for the SLAM state estimation implying that the map remains unchanged. A simple counter example is the case of a moving door in an indoor environment which changes the map of the scene.

A. Known Decomposition of the World

Object SLAM+ Object Tracking Interacting multiple models

In feature based mapping, motion of each feature can be assumed to be independent given the location of the feature at previous time step. In dense mapping, a scene/map be decomposed into n different parts such as chair, door etc. whose shape is known. The parts of the scene $m_k = \{b_k^i\}, 1 \leq i \leq n$ are assumed to move independently and hence the motion of the map can be represented as collection of independent motion of the parts. The true motion model for the each part of the scene is assumed to be one of the motion models $C \in \{C_j\}_{j=1}^{p}$ as represented in Section III.

Dropping the notation for scene part, In current formulation, we assume a uniform prior $\mu_j(0) = P(C_j), \sum_{j=1}^{p} \mu_j(0) = 1$ over different motion models for each scene part. However, this prior can be modified appropriately by object detection such as doors are more likely to have revolute joints etc.. Motion model probability is updated as more and more observations are received [11] as

$$\begin{aligned} \mu_j(k) &\equiv P(C_j|\mathbf{Z}_{0:k}) = \frac{P(z_k|\mathbf{Z}_{0:k-1}, C_j)P(C_j|\mathbf{Z}_{0:k-1})}{P(z_k|\mathbf{Z}_{0:k-1})} \\ \mu_j(k) &= \frac{P(z_k|\mathbf{Z}_{0:k-1}, C_j)\mu_j(k-1)}{\sum_{j=1}^{p} P(z_k|\mathbf{Z}_{0:k-1}, C_j)\mu_j(k-1)} \end{aligned} \tag{14}$$

The probability of the current observation z_k at time step k, conditioned over a specific motion model and all the previous observation can be represented by various method. In the current work, we filter the states using Extended Kalman Filter, for which this probability is the probability of observation residual w.r.t a normal distribution distributed with zero mean and innovation covariance [11].

B. No Prior Information

VIII. Temporal Structure

Articulation estimation provides us with better structure for the motion estimation problem however one still needs to estimate the evolution of parameters of these models over time e.g: Position of the object along an axis for prismatic joint. Temporal propagation of articulated bodies will require knowledge of dynamics model parameters (mass, friction etc.) apart from the external excitation (motor torque, force) applied to the system. Several approaches have been proposed for estimating these parameters that use the knowledge of some ground truth trajectories to estimate inertial and friction parameters [**?**] but they assume apriori access to the object. Furthermore, the external excitation can not be predicted as it can vary depending on the intention of agents.

The goal of our approach is to enforce a structure on the evolution of articulated motion without using any prior information specific to the current articulated body. We take our inspiration from neuroscience literature which posits that humans produce trajectories that are smooth in nature [**?**] to plan movements from one point to another point in environment. This smoothness assumption can be leveraged by using motion models that use only limited number of derivatives. To concertize, lets assume that x is the articulated model parameter (extension of a prismatic joint, angle of door along a hinge). The system model for a finite order motion model in continuous time domain with $\mathbb{X}(t) = [x, x^1, ., ., x^{n-1}]$ as the state can be written as

$$\begin{bmatrix} x^1 \\ x^2 \\ . \\ . \\ x^n \end{bmatrix} = \begin{bmatrix} 0 & 1 & . & . & 0 \\ 0 & 0 & . & . & 0 \\ 0 & 0 & . & . & 0 \\ 0 & 0 & . & . & 1 \\ 0 & 0 & . & . & 0 \end{bmatrix} \begin{bmatrix} x \\ x^1 \\ . \\ . \\ x^{n-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ 1 \end{bmatrix} \eta \tag{15}$$

where x^n denotes n^{th} order derivative of the state variable and η is the noise. This state propagation model can be converted to discrete time model as

$$\mathbb{X}(t + \delta t) = A\mathbb{X}(t) + B\eta \tag{16}$$

$$A = \begin{bmatrix} 1 & \delta t & \frac{\delta t^2}{2} & . & . \\ 0 & 1 & \delta t & . & . \\ 0 & 0 & 1 & . & . \\ 0 & 0 & 1 & . & . \\ 0 & 0 & . & . & . \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

where A is simply matrix exponential $\exp\{A^c \delta t\}$ of the matrix representation A^c in continuous time equation.

Following the [**?**], we assume jerk to be the noise in the system models. Using this method of providing temporal structure to the motion, we can write the motion of a prismatic joint with state $\mathbb{X}(k) = [x[k], y[k], v[k], \dot{v}[k], \ddot{v}[k], \dddot{v}[k]]$ as

$$\mathbb{X}(k+1) = \begin{bmatrix} 1 & 0 & \cos\theta & 0 & 0 \\ 0 & 1 & \sin\theta & 0 & 0 \\ 0 & 0 & 1 & \delta t & \frac{\delta t^2}{2} \\ 0 & 0 & 0 & 1 & \delta t \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbb{X}(k) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \eta \tag{18}$$

where θ is the direction of prismatic axis in 2D. The covariance of noise η in jerk needs to be continuously updated [**?**] to enable one to track all the possible range of smooth motions that can be performed by an articulated object.

IX. Results

References

[1] Shuang Wu, Hongjing Lu, and Alan L Yuille. Model selection and velocity estimation using novel priors for motion patterns. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances*

in Neural Information Processing Systems 21, pages 1793–1800. Curran Associates, Inc., 2009.

[2] Shimon Ullman. *The interpretation of visual motion.* MIT Press, 1979.

[3] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[4] João Paulo Costeira and Takeo Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[5] Jingyu Yan and M. Pollefeys. Automatic kinematic chain building from feature trajectories of articulated objects. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 712–719, June 2006.

[6] Roberto Martin Martin and Oliver Brock. Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2494–2501. IEEE, 2014.

[7] Dov Katz, Moslem Kazemi, J. Andrew (Drew) Bagnell, and Anthony (Tony) Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *Proceedings of IEEE International Conference on Robotics and Automation*, May 2013.

[8] Jürgen Sturm, Kurt Konolige, Cyrill Stachniss, and Wolfram Burgard. 3d pose estimation, tracking and model learning of articulated objects from dense depth video using projected texture stereo. In *Robotics: science and systems*, volume 2010, 2010.

[9] Xiaoxia Huang, Ian Walker, and Stan Birchfield. Occlusion-aware reconstruction and manipulation of 3d articulated objects. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1365–1371. IEEE, 2012.

[10] Sudeep Pillai, Matthew Walter, and Seth Teller. Learning articulated motions from visual demonstration. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

[11] Bar-Shalom Yaakov, XR Li, and Kirubarajan Thiagalingam. Estimation with applications to tracking and navigation. *New York: Johh Wiley and Sons*, 245, 2001.

[12] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006.

[13] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

[14] Tim Bailey. *Mobile robot localisation and mapping in extensive outdoor environments*. PhD thesis, Citeseer, 2002.

[15] Chieh-Chih Wang, Charles Thorpe, and Sebastian Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 842–849. IEEE, 2003.