2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

QSRNet: Estimating Qualitative Spatial Representations from RGB-D Images

Sang Uk Lee Sungkweon Hong Andreas Hofmann Brian Williams

***Abstract*— Humans perceive and describe their surroundings with qualitative statements (e.g., "Alice's hand is in contact with a bottle."), rather than quantitative values (e.g., 6-D poses of Alice's hand and a bottle). Qualitative spatial representation (QSR) is a framework that represents the spatial information of objects in a qualitative manner. Region connection calculus (RCC), qualitative trajectory calculus (QTC), and qualitative distance calculus (QDC) are some popular QSR calculi. With the recent development of computer vision, it is important to compute QSR calculi from the visual inputs (e.g., RGB-D images). In fact, many QSR application domains (e.g., human activity recognition (HAR) in robotics) involve visual inputs. We propose a qualitative spatial representation network (QSRNet) that computes the three QSR calculi (i.e., RCC, QTC, and QDC) from the RGB-D images. QSRNet has the following novel contributions. First, QSRNet models the dependencies among the three QSR calculi. We introduce the dependencies as *kinematics for QSR* because they are analogous to the kinematics in classical mechanics. Second, QSRNet applies the 3-D point cloud instance segmentation to compute the QSR calculi. The experimental results show that QSRNet improves the accuracy in comparison to the other state-of-the-art techniques.**

I. Introduction

Humans can perceive and discuss their surroundings effectively. Although the secrets for human cognition have not been discovered, many researchers agree that the ability to understand the environment on an abstract and qualitative level plays an important role [1]. For example, in recognizing Alice's activity of picking up a bottle, the qualitative statement, "Alice's hand is in contact with a bottle," would be more compact and effective than reasoning using the quantitative 6-D poses (i.e., x, y, z, $roll$, $pitch$, and yaw) of Alice's hand and the bottle. In fact, within the field of artificial intelligence (AI), qualitative representation and reasoning has been developed as a crucial framework.

Qualitative spatial representation (QSR) is a framework for representing spatial information about objects in a qualitative manner (e.g., a qualitative relation of, "Alice's hand is not in contact with a bottle."). QSR has many applications, including geographic science and human activity recognition (HAR) [2]. In particular, HAR is a very interesting domain for QSR. This is because an accurate HAR system is required when humans interact with various objects in a small workspace. HAR is the study of how to make robots recognize which activity a human is doing (e.g., recognizing the activity of, "Alice is picking up a bottle."). It is a crucial ingredient for a successful human–robot collaboration. Many HAR studies have used QSR to represent key predicates (e.g., "Alice is holding a bottle.") and human activities (e.g., "Alice is picking up a bottle.") [3]–[5]. In fact, we used HAR as the main example domain for our work.

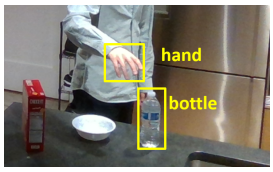
Several different QSR calculi have been used for HAR. Different QSR calculi are specialized for representing the different qualitative relations. In terms of the popular QSR calculi, region connection calculus (RCC) [2] has been proven to be effective in qualitatively representing mereotopological information among the objects (e.g., "A human hand is disconnected from a bottle," and, "The bottle is in a refrigerator."). Qualitative trajectory calculus (QTC) [6] has been used in representing relative motions among the objects (e.g., "A human hand is moving toward the bottle."). Qualitative distance calculus (QDC) is useful in representing how near or far objects are [5].

When applying QSR, computing the calculi from visual inputs (e.g., RGB-D images) is key. This is because many recent applications involve visual inputs. This paper introduces the qualitative spatial representation network (QSRNet), which computes the above three popular QSR calculi (i.e., RCC, QTC, and QDC) from the RGB-D images. QSRNet has a layered structure that is composed of a neural network and a dynamic Bayesian network (DBN). The neural network applies the instance segmentation to obtain 3-D point cloud instance masks of the objects (see Figure 1(a)) and computes several key metrics. The DBN captures useful dependencies among the different calculi and computes QSR relations from the key metrics. The dependencies have an analogy to the kinematics in classical mechanics. Thus, we introduce the dependencies as *kinematics for QSR*.

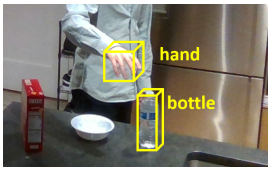
Our work has the following novel contributions. First, we introduce the kinematics for QSR that models the dependencies among the different QSR calculi. In the QSR community, different calculi have been independently studied and are specialized for representing specific qualitative relations. Thus, their dependencies have not been considered before. Even a well-used software library called QSRlib [1] computes the various calculi independently and it does not consider the dependencies. However, we emphasize that there are useful dependencies among the different QSR calculi. For example, let us consider two calculi, RCC and QTC. If, "The hand is not in contact with the bottle," (an RCC relation) and, "The hand is moving toward the bottle," (a QTC relation), it is reasonable to think that, "The hand will be in contact with the bottle," (another RCC relation) after some time. By making the analogies that RCC corresponds to the displacement and QTC corresponds to the velocity,

The authors are with the MIT CSAIL, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA (e-mail: sangukbo@mit.edu, sk5050@mit.edu, hofma@mit.edu, williams@mit.edu)

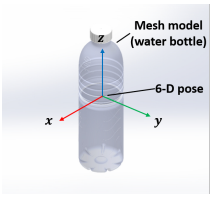
(a) 3-D point cloud instance masks of a hand and a bottle



(b) 2-D bounding boxes



(c) 3-D bounding boxes



(d) 3-D mesh model around a bottle's 6-D pose

Fig. 1: Different ways to compute the QSR relations

we can model a kinematic relation between RCC and QTC. A similar relation can be modeled for QDC and QTC. Furthermore, we can formulate a DBN (which corresponds to the DBN part of QSRNet) from the kinematic relations. We show that our combined model, which considers the kinematics for QSR, is more accurate than the previous models that ignored the dependencies among the different QSR calculi.

Second, QSRNet uses the 3-D point cloud instance masks of the objects to compute the qualitative relations. The 3-D point cloud instance segmentation (i.e., finding the 3-D point cloud instance masks of the objects, see Figure 1(a)) is a relatively new research topic. We introduce that it can be very effective for computing the various QSR relations. Let us consider how the previous works computed the QSR relations. [7] used 2-D bounding boxes (see Figure 1(b)) and [1], [4] used 3-D bounding boxes (see Figure 1(c)) of objects to acquire the QSR relations (e.g., "A hand is in contact with a bottle," if the bounding boxes for the hand and a bottle overlap). The bounding boxes can be very inaccurate for computing QSR calculi, especially for objects with complex shapes. To resolve this, [3] first computed 6-D poses of the objects and acquired the QSR relations by placing accurate 3-D mesh models of the objects around the 6-D poses (see Figure 1(d)). However, the mesh models need to be constructed a priori. Thus, this approach is not suitable when we encounter objects of which we do not have the mesh models. We emphasize that 3-D point cloud masks can capture the accurate shape of objects and the instance segmentation to compute the masks does not require a priori construction of the object models.

This paper is organized as follows. Section II provides the related background for this research. A detailed illustration of the kinematics for QSR is presented in Section III. Section IV presents QSRNet. The experimental evaluations are provided in Section V. Finally, Section VI concludes the paper.

II. Background

A. Region Connection Calculus (RCC)

RCC is very useful in capturing mereotopological relations between the objects (e.g., "A human hand is in contact with a bottle."). In RCC, there are a finite number of possible qualitative relations between any two given objects or regions, A and B, in $\mathbb{R}^3$ space [2]. The RCC relations take the objects' geometric shapes into account, and it is very effective in capturing the qualitative relations between the objects of complex shapes (e.g., human hand, wine glass).

In our experiment, we used RCC-5, which has five possible qualitative relations, as follows: A is disconnected from B ($DC(A, B)$) (i); A is partially occluded by B ($PO(A, B)$) (ii); A is identical to B ($EQ(A, B)$) (iii); A is a proper part of B, or the inverse ($PP(A, B)$ or $PPi(A, B)$) (iv and v). Figure 2 visualizes the five relations.

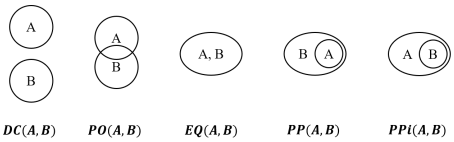
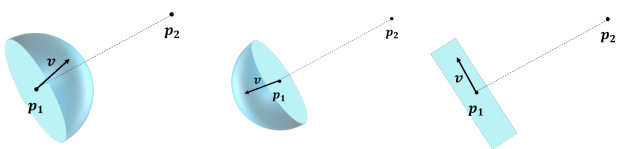


Fig. 2: RCC-5 relations.

B. Qualitative Trajectory Calculus (QTC)

QTC can represent the relative motions qualitatively (e.g., "A human hand and a bottle are moving toward each other.") [6]. Unlike RCC, QTC is defined for the representative points of objects (e.g., the centers of a hand and a bottle) and does not include the complete geometry of the object. Thus, the QTC statement, "A hand is moving toward a bottle," would actually mean, "The center of a hand is moving toward the center of a bottle." Note that this was not the case for RCC because it considers the entire geometry of the object.

Let us assume that we are given two points, p_1 and p_2, where p_1 is the center of a hand and p_2 is the center of a bottle. When p_2 is fixed, there can be three possible movements for p_1: i) p_1 is moving toward p_2 (represented as – state), ii) p_1 is moving away from p_2 (represented as + state), and iii) p_1 is neither moving toward nor away from p_2 (represented as 0 state). Figure 3 visualizes the three movements. As we consider relative motions in a 3-D space, any velocity vector on the half spheres (or plane) in Figure 3 corresponds to the same relation.



(a) – state (b) + state (c) 0 state

Fig. 3: Three possible movements of p_1, assuming p_2 is fixed.

When both p_1 and p_2 are moving, we can represent the relative motion with a 2-D tuple (e.g., (+, +)). The first entry

represents p_1's relative motion while assuming that p_2 is fixed at the current location for some particular length of time (we denote this fixed point as p_2^{fixed}). Thus, the first entry represents p_1's motion with respect to p_2^{fixed}. The second entry represents p_2's relative motion assuming that p_1 is fixed (i.e., p_2's motion with respect to p_1^{fixed}).

In QTC, we can append other entries to the 2-D tuple from above to represent other aspects of the relative motions. In this paper, we represent QTC relations with a 3-D tuple. The first two entries are the same as above. The last entry represents how the distance between p_1 and p_2 changes, without assuming one of them is fixed. Let $d(t)$ be the distance between p_1 and p_2 at time $t \in [0, \infty)$. Then, the third entry has + state if $\frac{d}{dt}d(t) > 0$ (i.e., p_1 and p_2 are moving away from each other), – state if $\frac{d}{dt}d(t) < 0$ (i.e., moving toward each other), and 0 state if $\frac{d}{dt}d(t) = 0$ (i.e., neither moving toward nor away from each other). Note that for the third entry, we do not assume one of the two points is fixed, as we did for the first and second entries. We are considering the rate of the distance between the two points, which are moving freely. Thus, the third entry conveys different information from the first and second entries.

C. Qualitative Distance Calculus (QDC)

QDC can qualitatively represent how far the two objects are apart [1], [5]. Similar to QTC, QDC is defined by the representative points of the objects (e.g., the centers of a hand and a bottle). QDC expresses the qualitative distance relations between the two points depending on the defined region boundaries. For example, let us allow five qualitative relations of *very close*, *close*, *commensurate*, *far*, and *very far*. Then, we can define two points as *very close* if their distance is less than 0.5 m, *close* if the distance is greater than or equal to 0.5 m but less than 1 m, and so on. The QDC used in this paper is shown in Figure 4. Note that the distance numerics in Figure 4 can be chosen arbitrarily by users.

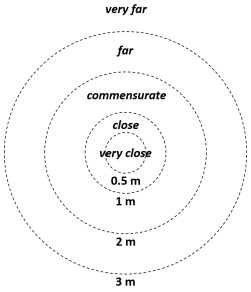


Fig. 4: An example of QDC with five qualitative relations

Even though QDC conveys a very simple concept of distance relations with the regional boundaries, it is very useful when it is paired with RCC [1]. When two objects are in the DC relation in terms of RCC, RCC cannot provide any information about how far apart they are. However, this information can be useful in many cases. For example, even though RCC tells us a hand and a bottle are disconnected, we might want to have some idea of how near or far apart they are from each other. By using QDC, we can qualitatively represent how far apart they are.

III. Kinematics for Qualitative Spatial Representations (QSR)

We first illustrate the kinematics for QSR. In this paper, We consider the kinematic dependencies among the three popular QSR calculi: RCC, QTC, and QDC. We make one remark first. QTC and QDC are qualitative relations referring to representative points of the objects. We will consider the geometric centers of the objects. We do not need such specification for RCC.

The following well-known kinematic equation is from classical mechanics. It shows how to relate the displacement and the velocity.

$$x(t_2) = x(t_1) + \int_{t_1}^{t_2} v(t)dt \tag{1}$$

$x(t)$ is the displacement and $v(t)$ is the velocity at time t.

Note that for the discrete-time steps, the following equation is obtained:

$$x(T_{k+1}) = x(T_k) + \int_{T_k}^{T_{k+1}} v(t)dt \simeq x(T_k) + \Delta T \times v(T_k) \tag{2}$$

T_k and T_{k+1} are discrete time steps. $\Delta T = T_{k+1} - T_k$ is the unit interval for the discrete time steps. In the last approximation, we made an assumption that $v(t) \simeq v(T_k)$ for all t in $[T_k, T_{k+1}]$.

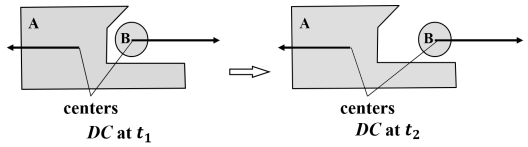
The main idea of the kinematics for QSR is that RCC and QDC represent the positional relations that are analogous to the displacement. In contrast, QTC represents the motions that are analogous to the velocity. For example, the RCC statement $DC(hand, bottle)$ represents a static and a positional relation that, "The $hand$ and the $bottle$ are not in contact." A QDC statement $close(hand, bottle)$ also represents a static and positional relation that, "The two are close". A QTC statement $((-, 0, -), hand, bottle)$ represents a dynamic motion that, "The $hand$ is moving toward the $bottle$." Thus, we can come up with equations that are similar to Eq. (1) and Eq. (2). This is why we introduce our idea as *kinematics*.

Now, we elaborate on how to model the kinematics. We model two types of kinematic relations: i) the kinematics between RCC and QTC and ii) the kinematics between QDC and QTC. To be more specific, we model two discrete-time stochastic processes that correspond to Eq. (2). The discrete time models can easily be represented as a DBN model. Let us first consider the kinematics between RCC and QTC. For example, if, "The hand is not in contact with the bottle," (i.e., in the DC relation) but, "The hand is moving toward the bottle," (i.e., in the (–, 0, –) relation), it is reasonable to think that, "The hand will be in contact with the bottle," at some time. One caveat is that such a

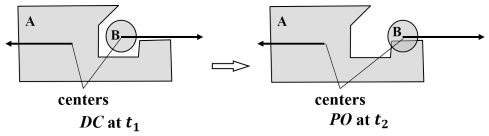
transition is not deterministic. For example, let us consider the case in Figure 5 where, "$object_A$ is not in contact with $object_B$," initially at time t_1. If "$object_A$ is moving away from $object_B$," it is probable that, "$object_A$ is going to remain to be not in contact with $object_B$," as in Figure 5(a). However, there is also a chance, maybe with less probability, that the two objects might overlap (i.e., in the PO relation) depending on the geometries of the objects as shown in Figure 5(b). Thus, the example shows that the kinematics that relate RCC and QTC should be modeled stochastically, rather than deterministically as it was in classical mechanics. The following stochastic process models the kinematics between RCC and QTC, and it corresponds to Eq. (2) in classical kinematics.

$$R(T_{k+1}) \simeq I_R(R(T_k), Q(T_k), W_R(T_k)) \tag{3}$$

I_R is an operator that corresponds to the integral (i.e., $\int_{T_k}^{T_{k+1}} \cdot \, dt$) in Eq. (2). $R(T_k)$ is the RCC relation, and $Q(T_k)$ is the QTC relation at the time step T_k. Here, we made an assumption that $Q(t)$ remains as $Q(T_k)$ for all t in $[T_k, T_{k+1}]$. $W_R(T_k)$ governs the stochasticity of the process.



(a) Two objects move away and remain DC.



(b) Two objects move away but become PO.

Fig. 5: An example of non-deterministic dependency between RCC and QTC relations (note: $t_2 > t_1$).

The kinematics between QDC and QTC can be modeled in a similar way. For example, if there are two objects that are moving away from each other (i.e., a QTC relation), it is also reasonable to guess that the two objects are going to be further apart (i.e., a QDC relation). This can also be modeled as a stochastic process as shown.

$$D(T_{k+1}) \simeq I_D(D(T_k), Q(T_k), W_D(T_k)) \tag{4}$$

I_D is also an integral-like operator (different from I_R in Eq. (3)). $D(T_k)$ is the QDC relation at the time step T_k. $W_D(T_k)$ governs the stochasticity of the process.

In summary, we have modeled the kinematics among the three QSR calculi with the two stochastic processes:

- The stochastic process in Eq. (3) models the kinematic relations between RCC and QTC.
- The stochastic process in Eq. (4) models the kinematic relations between QDC and QTC.

We combined three different QSR calculi, which have been independently developed. We can model more dependencies among the three QSR calculi other than the kinematic dependencies. For example, by considering RCC and QTC, it is reasonable to think that the motion of an object is more likely to change when a human interact with it. Moreover, the interaction is possible when the object is not disconnected from the human (i.e., $\neg DC$). Thus, it is reasonable to say that the QTC relation between the object and the human is more likely to change when the object is not disconnected from the human. This example models our intuition on the dynamics of how the motions of objects would change (i.e., by interacting with a human). This is not included in the current version of QSRNet, since we focus on modeling the kinematic relations among the QSR calculi.

IV. Qualitative Spatial Representation Network (QSRNet)

In this section, we illustrate QSRNet, which works in three steps. The first step is to perform 3-D point cloud instance segmentation and to compute accurate 3-D point cloud masks of the objects. The second step is to compute some key metrics using the 3-D point cloud masks. The third step is to compute the QSR relations from the computed key metrics. The overall architecture of QSRNet is visualized in Figure 6. The following subsections describe each of these three steps.

Note that RCC, QTC, and QDC represent the qualitative relations between a pair of objects. We explain QSRNet with an example pair of objects (e.g., a hand and bottle pair). If there are more than two pairs of objects, QSRNet can compute the RCC, QTC, and QDC relations for each object pair (e.g., hand and bottle pair, hand and bowl pair). In this case, we construct 3-D point cloud masks for all of the objects in the first step. In the second step, we then compute the metrics for each pair. In the third step, we compute the QSR relations for each pair by using the metrics.

A. Instance Segmentation

In the first step, QSRNet computes the 3-D point cloud masks of the objects (Figure 7(b)). The inputs are the RGB-D images. Figure 7 shows an example input and output of the first step. Figure 7(b) zooms in for a hand and a bottle.

We determine the 3-D point cloud masks as follows. First, we compute the 2-D instance masks of the objects over the RGB images only (not using depth images yet). We can use any 2-D instance segmentation algorithm over the RGB images, such as Mask R-CNN [8]. Next, for each of the 2-D masks in the RGB images, we generate a 3-D point cloud mask using the depth images. That is, for every pixel in the 2-D mask, we use the corresponding depth value from the depth images to generate a point in a 3-D space using Eq. (5).[1]

$$x = \frac{(u - c_x) \times d}{f_x}, \quad y = \frac{(v - c_y) \times d}{f_y}, \quad z = d \tag{5}$$

[1]This is the standard method of generating 3-D point clouds from RGB-D images.

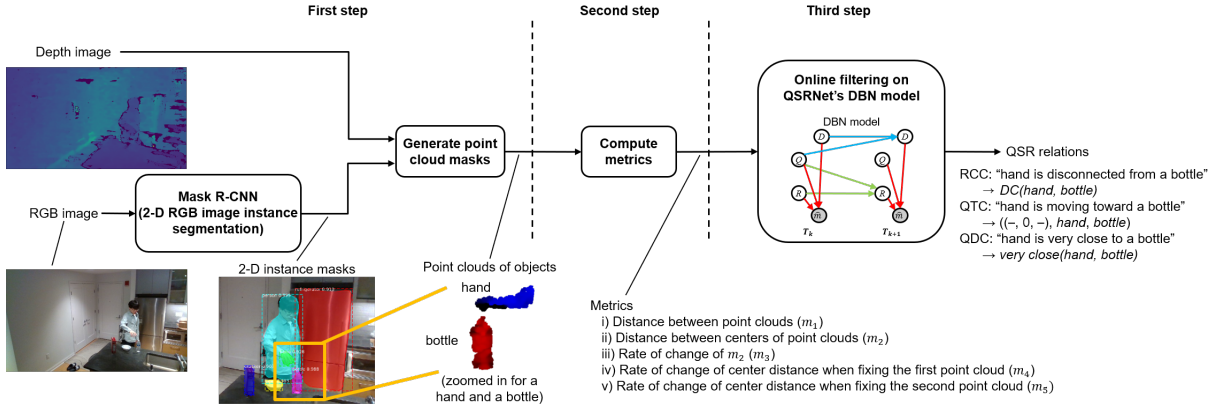
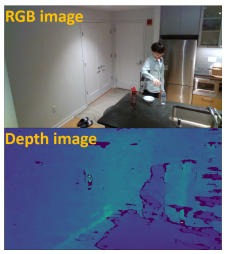
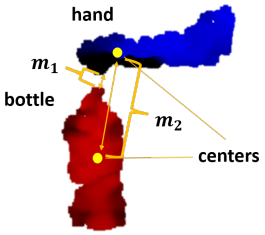


Fig. 6: QSRNet architecture



(a) An example input.



(b) An example output masks of a human hand and a bottle. Two metrics (m_1 and m_2) from the second step are superposed (zoomed in view).

Fig. 7: Input and output for the instance segmentation step.

In Eq. (5), x, y, and z are the coordinates in a 3-D space with respect to the camera frame. u and v are the coordinates of the pixel in the depth image. c_x, c_y, f_x, and f_y are the intrinsic parameters of the depth camera.

If we use multiple RGB-D cameras from multiple viewpoints, we can stitch together the object's 3-D point cloud masks from all of the cameras. Using multiple cameras would make the 3-D point cloud masks (and also QSRNet) more accurate. In addition, it can be helpful in scenarios where visual occlusion (e.g., objects occluding each other) is critical. However, we would like to emphasize that QSRNet performs well even with a single RGB-D camera in most cases. We used only one RGB-D camera for our experiment.

We could have used a different method for computing the 3-D point cloud masks. For example, we could first construct a 3-D point cloud map from the RGB-D images and then apply an instance segmentation algorithm in [9], which computes the 3-D point cloud masks of the objects directly from a 3-D point cloud map. However, the direct instance segmentation on a 3-D point cloud map is a relatively new research topic, and it therefore often lacks labeled training data for various robotics application domains including HAR. The 2-D instance segmentation (i.e., computing the 2-D instance masks over the RGB images), on the other hand, is well-researched. It has many datasets available such as Microsoft COCO dataset (for various objects) [10] and DensePose dataset (for human body parts) [11]. Thus, we emphasize that the method introduced in this subsection (i.e., using the 2-D instance segmentation for computing the 3-D point cloud instance masks) can be effective.

B. Computing Metrics

In the second step, we compute five key metrics from the objects' 3-D point cloud masks. Figure 7(b) shows some of the metrics with the masks of a human hand and a bottle. The first metric (m_1) is the distance between the masks. Eq. (6) formally presents the distance. Here, $dist(\cdot)$ refers to the distance between two points in a 3-D space. M_1 and M_2 refer to two different point cloud masks (e.g., human hand and bottle). Note that the masks consist of points and they are small point clouds. p_1 and p_2 are the points in the masks.

$$m_1 = min(\{dist(p_1, p_2) |\ p_1 \in M_1 \text{ and } p_2 \in M_2\}) \quad (6)$$

The second metric (m_2) is the distance between the centers of the two point cloud masks. The third metric (m_3) is the rate of change of m_2; m_2 and m_3 are given by Eq. (7). Here, c_1 and c_2 refer to the geometric centers of the two masks M_1 and M_2 (i.e., c_1 is the average point of M_1).

$$m_2 = dist(c_1, c_2), \quad m_3 = \frac{d}{dt} m_2 \quad (7)$$

The fourth metric (m_4) is the rate of change of the distance between the centers of the two masks, assuming that the center of the first mask has been fixed for a short amount of time. The fifth metric (m_5) is the similar rate of change except that we fix the center of the second mask. m_4 and m_5 are given by Eq. (8). Here, c_1^{fixed} indicates the center of the first mask, assuming it has been fixed for some time.

$$m_4 = \frac{d}{dt} dist(c_1^{fixed}, c_2), \quad m_5 = \frac{d}{dt} dist(c_1, c_2^{fixed}) \quad (8)$$

Note that all five metrics are for a pair of objects. Thus, if we have multiple pairs of interest (e.g., hand and bottle pair, hand and bowl pair), we compute the metrics for each pair.

C. Computing QSR Relations

The third step computes the QSR calculi using the metrics from the previous step. The third step is designed as a DBN. The DBN models the kinematics for QSR. This is given by the two discrete-time stochastic processes in Eq. (3) and Eq. (4). We present the DBN model in Figure 8.

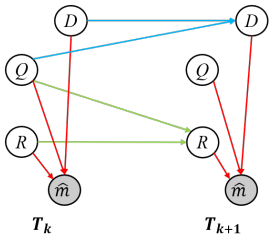


Fig. 8: DBN that models the kinematics for QSR.

In Figure 8, the nodes named R, Q, and D represent the hidden states for the three QSR calculi relations. R is for RCC, Q is for QTC, and D is for QDC. R, Q, and D at the time step T_k correspond to the $R(T_k)$, $Q(T_k)$, $D(T_k)$ in Eq. (3) and Eq. (4), respectively. The gray nodes represent the observations. For the observations, we use the metrics computed in the second step of QSRNet.

There are three types of conditional probability distributions in DBN, which are indicated with red, green, and blue edges, respectively. The red edges represent the observation models (representing $P(\hat{m}(T_k)|R(T_k), Q(T_k), D(T_k))$). Here, $\hat{m}(T_k) = [m_1(T_k), m_2(T_k), m_3(T_k), m_4(T_k), m_5(T_k)]$ is a vector of the five metrics at the time step T_k. In this paper, we assume the observation model follows a multivariate normal distribution (i.e., $\hat{m}(T_k) \sim \mathcal{N}(\mu(R(T_k), Q(T_k), D(T_k)), \Sigma(R(T_k), Q(T_k), D(T_k)))$).

Let us consider the other conditional probability distributions that are indicated in green and blue. The green edges represent $P(R(T_{k+1})|R(T_k), Q(T_k))$, and they model the kinematics between RCC and QTC given by Eq. (3). The stochastic process in Eq. (3) says that $R(T_{k+1})$ would depend on $R(T_k)$ and $Q(T_k)$. It directly corresponds to $P(R(T_{k+1})|R(T_k), Q(T_k))$. The blue edges represent $P(D(T_{k+1})|D(T_k), Q(T_k))$, and they model the kinematics between QDC and QTC given by Eq. (4) in a similar manner.

To compute the QSR relations online, we can perform online filtering over the DBN with the observations computed in the second step of the QSRNet. We can apply any of the DBN online filtering algorithms from [12]. We refer to [12] for more information on the DBN online filtering.

We make several remarks. First, the conditional probability distributions can be learned from the data, if necessary. We refer to [12] for the details on how to learn the conditional probability distributions, as that is not the focus of this paper. Second, the DBN model in Figure 8 is for the QSR calculi between one pair of objects. If we want to compute the QSR calculi for more than one pair, we can construct a DBN model for each pair.

V. Evaluations

A. Experimental Setting

For the experimental evaluations, we collected video streams, where each video contains a human interacting with various objects in a kitchen environment. The experimental environment is shown in Figure 9. There are six objects: a refrigerator, a dining table, a bottle of water, a bowl, an apple, and a box of crackers. The human interacts with the objects by performing five activities: open, close, pick, place, and approach. The human could open or close the refrigerator; pick up or place down the bottle of water, the bowl, the apple, and the box of crackers; or approach the refrigerator or the dining table. Note that these five types of activities are the fundamental ones that have been used in previous studies. For example, [1], [3] computed QSR relations while a human performed opening, closing, picking, and placing activities with five different objects. The Opportunity Activity Recognition dataset used in [13] mostly consists of opening and closing seven different objects.

Fig. 9: The experimental environment.

We collected more than 20 video streams and each video was about 30 seconds in length. In each video, a human performed activities in a random sequence. For example, one of the videos contained a human performing the following activities in a sequence: i) opening the refrigerator, ii) picking up the bowl from the refrigerator, iii) placing the bowl down on the dining table, iv) picking up the bottle, v) placing the bottle in the refrigerator, and vi) closing the refrigerator. For the videos, a human supervisor labeled the ground truth QSR relations of object pairs.

Figure 10 shows four frames that were extracted from a video while a human is picking up the water bottle. The sequence of the frames are numbered. As one can see, the objects were densely collocated to make the experiment more interesting. We computed the RCC, QTC, and QDC relations for the pairs listed in Table I; a total of 20 pairs were tracked. The recognition was done with a frequency of 4 Hz.

TABLE I: Object Pairs Estimated for QSR Calculi

Object Pairs	Object Pairs	Object Pairs
human (torso), refrigerator	human, dining table	human, bottle
human, bowl	human, apple	human, cracker
hand, refrigerator	hand, dining table	hand, bottle
hand, bowl	hand, apple	hand, cracker
refrigerator, bottle	refrigerator, bowl	refrigerator, apple
refrigerator, cracker	dining table, bottle	dining table, bowl
dining table, apple	dining table, cracker	-

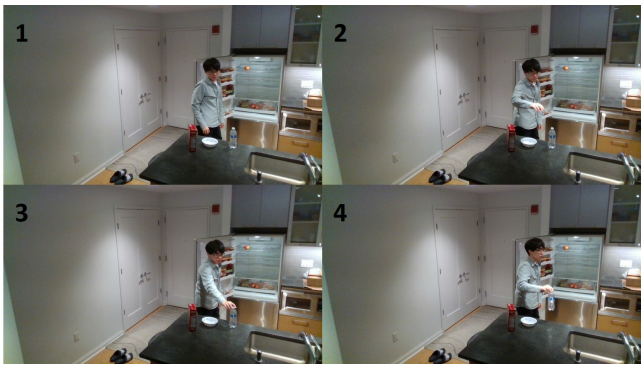


Fig. 10: Four frames from a video.

B. Evaluation of QSRNet

1) QSRNet Comparison: We provide the experimental results for QSRNet. First, we compare QSRNet with three other studies, which are briefly introduced. [3] computed the 6-D poses, then applied accurate 3-D mesh models of the objects to compute the QSR relations. We refer to [3] as the "mesh model" method.[2] [7] used 2-D bounding boxes of the objects to compute the QSR relations. We refer to [7] as the "2-D bounding box" method. "QSRlib" in [1] used 3-D bounding boxes to compute the QSR relations.[3]

Before proceeding, let us briefly explain how the accuracy rates in Table II (and all of the other tables) were computed. For each calculi, we first computed the accuracy for each pair of objects in Table I. For each pair, the accuracy was a ratio between the total number of correct recognitions and the total number of samples (i.e., total number of time steps). For example, if we got correct recognition for 8,000 out of 10,000 samples (i.e., time steps), the accuracy would be 8,000/10,000$\times$100 = 80%. Then, for each calculi, Table II shows the accuracies that are averaged over all pairs.

Table II compares the accuracies when applying QSRNet and the other three studies for this experiment. For all three calculi, the 2-D bounding box method shows the worst accuracy because the 2-D bounding boxes fail to capture the 3-D information about the objects. For example, when computing the RCC relations, two objects in a 3-D space might not be in contact, even if the 2-D bounding boxes in the RGB image overlap. In this case, the 2-D bounding box method would incorrectly estimate that the two objects are in the *PO* relation. Conversely, QSRlib uses 3-D bounding boxes; hence, it has a better accuracy. However, 3-D bounding boxes fail to capture the complex shapes of objects. 3-D point cloud masks (used in QSRNet) or 3-D mesh models (used in the mesh model method) can capture the detailed shapes of objects more accurately. In fact, QSRNet and the mesh model method had the best performance. In addition, they show comparable accuracies for the three calculi.

Although QSRNet and the mesh model method performed similarly well in Table II, the mesh model method has a significant limitation. The mesh model method requires a priori construction of the mesh models. Needless to say, the construction of the mesh models can be a tedious job. Furthermore, the mesh model method would not be able to compute the QSR calculi correctly if the mesh models were not constructed. A scenario can easily be designed where this limitation is critical. For example, let us consider the case where a human is interacting with a water bottle, a bowl, an apple, and a box of crackers of different shapes and sizes from the ones in Figure 11(a) (see Figure 11(b)). The objects in Figure 11(a) were used to get the results in Table II. We did not construct the mesh models for the new objects in Figure 11(b); we only constructed the mesh models for the objects shown in Figure 11(a). Even though the human interacts with a new water bottle, the recognition system needs to tell us that, "A human hand is in contact with a bottle," when the human is holding the bottle. QSRNet has no problem performing this task. However, the mesh model method can be highly inaccurate because the mesh models we have are different from the new objects.

TABLE II: Accuracy Comparisons

Different Estimators	Accuracy		
	RCC	QTC	QDC
QSRNet	**94.82%**	90.65%	97.88%
Mesh model	93.16%	**91.03%**	**98.04%**
2-D bounding box	72.50%	71.69%	87.93%
QSRlib	86.91%	89.26%	95.49%

(a) Objects with mesh models constructed

(b) New objects without mesh models

Fig. 11: Two different sets of objects.

To verify this, the same experiment from above was performed with the new objects in Figure 11(b). Again, we did not construct the mesh models for the new objects and we only have the mesh models for the objects in Figure 11(a). Table III shows the accuracy rates. Compared to the results in Table II, Table III shows little change for QSRNet. However, the accuracy rates for the mesh model method decreased much as expected. Especially, the mesh model method had a hard time estimating the RCC relations correctly, because the RCC relations are affected the most by the objects' shape.

2) How Kinematics Helps QSR: QSRNet models the kinematics among the different QSR calculi. We verify that the modeling of the kinematics improves the accuracy. This was achieved by comparing QSRNet with a comparison model that does not use the kinematics.

[2] [3] computed the RCC relation only; however, we were able to use the computed 6-D poses to compute the QTC and QDC relations easily.

[3] The bounding boxes were calculated from the viewpoint of camera.

TABLE III: Accuracy Comparisons Using New Objects

Different Estimators	Accuracy		
	RCC	QTC	QDC
QSRNet	**93.57%**	**90.26%**	**98.03%**
Mesh model	82.96%	90.06%	96.42%
2-D bounding box	75.31%	70.32%	86.84%
QSRlib	84.29%	88.55%	94.78%

Let us first explain the comparison model. In QSRNet, the kinematics are modeled in the DBN. We can remove the modeling of the kinematics by replacing the DBN with three independent hidden Markov models (HMMs) (one for each of RCC, QTC, and QDC relations). The hidden variable of each of the HMMs represents the state of RCC, QTC, or QDC relations, respectively. We can perform online filtering over the HMMs, just like we do with the DBN model. The online observations for the independent HMMs are the same as that for the DBN model. In other words, the online observations for the independent HMMs are the metrics computed from the second step of QSRNet. Thus, the comparison model is the same as QSRNet except for the third step. In the third step, we replace the DBN with the three independent HMMs. Figure 12 visualizes one of the three independent HMMs in the comparison model. Though Figure 12 visualizes only one HMM, the comparison model has three independent copies of such HMM.

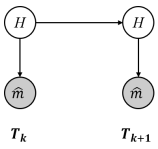


Fig. 12: An independent HMM (H represents one of the three QSR calculi and $\hat{m}$ represents the metrics).

Table IV provides the accuracy rates for applying QSRNet and the comparison model to our experiment. For all three calculi, the accuracies of QSRNet are better than the comparison model. This shows that the kinematics for QSR helps to improve the model accuracy.

TABLE IV: Accuracy Comparisons with the Comparison Model

Different Estimators	Accuracy		
	RCC	QTC	QDC
QSRNet	**94.82%**	**90.65%**	**97.88%**
Comparison model	90.97%	89.71%	96.10%

VI. Conclusions

We present QSRNet, a new architecture for computing the RCC, QTC, and QDC relations of the objects from RGB-D images. QSRNet works in three steps. The first step uses a neural network to construct the 3-D point cloud masks of the objects from the RGB-D images. The second step computes the key metrics using the 3-D point cloud masks. The third step performs online filtering on a DBN, which models the dependencies among the three QSR calculi (i.e., RCC, QTC, and QDC). We use the key metrics as observations in online filtering and compute the QSR relations. We validate QSRNet through an experimental kitchen scenario.

In this paper, we used three popular QSR calculi: RCC, QTC, and QDC. However, there are other useful QSR calculi as well. For instance, the cardinal direction relation [1] and ternary point configuration calculus (TPCC) [14] are good QSR calculi for representing directional information. Future efforts will focus on including other QSR calculi in QSRNet. The code for QSRNet is provided on https://github.com/sangukbo/qsrnet.

References

[1] Y. Gatsoulis, M. Alomari, C. Burbridge, C. Dondrup, P. Duckworth, P. Lightbody, M. Hanheide, N. Hawes, D. Hogg, A. Cohn, *et al.*, "QSRlib: a Software Library for Online Acquisition of Qualitative Spatial Relations from Video," 2016.

[2] A. G. Cohn, B. Bennett, J. Gooday, and N. M. Gotts, "Qualitative Spatial Representation and Reasoning with the Region Connection Calculus," *GeoInformatica*, vol. 1, no. 3, pp. 275–316, Oct. 1997.

[3] S. U. Lee, A. Hofmann, and B. Williams, "A Model-Based Human Activity Recognition for Human–Robot Collaboration," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*. Macau, China: IEEE, Nov. 2019.

[4] P. Duckworth, M. Alomari, Y. Gatsoulis, D. C. Hogg, and A. G. Cohn, "Unsupervised Activity Recognition Using Latent Semantic Analysis on a Mobile Robot," in *Proceedings of the Twenty-second European Conference on Artificial Intelligence*. IOS Press, 2016, pp. 1062–1070.

[5] E. Clementini, P. Di Felice, and D. Hernández, "Qualitative Representation of Positional Information," *Artificial intelligence*, vol. 95, no. 2, pp. 317–356, 1997.

[6] N. Van de Weghe, A. Cohn, G. De Tre, and P. De Maeyer, "A Qualitative Trajectory Calculus as a Basis for Representing Moving Objects in Geographical Information Systems," *Control and Cybernetics*, vol. 35, no. 1, pp. 97–119, 2006.

[7] M. Sridhar, A. G. Cohn, and D. C. Hogg, "From Video to RCC8: Exploiting a Distance Based Semantics to Stabilise the Interpretation of Mereotopological Relations," in *International Conference on Spatial Information Theory*. Springer, 2011, pp. 110–125.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[9] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas, "GSPN: Generative Shape Proposal Network for 3D Instance Segmentation in Point Cloud," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3947–3956.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[11] R. Alp Güler, N. Neverova, and I. Kokkinos, "DensePose: Dense Human Pose Estimation In The Wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7297–7306.

[12] K. P. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning," PhD Dissertation, University of California, Berkeley, 2002.

[13] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, *et al.*, "Collecting Complex Activity Datasets in Highly Rich Networked Sensor Environments," in *Networked Sensing Systems (INSS), 2010 Seventh International Conference on*. IEEE, 2010, pp. 233–240.

[14] R. Moratz, B. Nebel, and C. Freksa, "Qualitative Spatial Reasoning about Relative Position," in *International Conference on Spatial Cognition*. Springer, 2002, pp. 385–400.