Collision Avoidance in Depth Space

I. Introduction

When humans and robots share the same work space, safety is the primary issue of concern [8]. Secondary but not negligible is to prevent robot damages due to collisions with unforeseen obstacles. While potential injuries of unexpected human-robot impacts can be limited by lightweight/compliant mechanical designs of manipulators and collision reaction strategies [7], preventing collisions in a dynamic and largely unpredictable environment relies on the extensive use of exteroceptive sensors.

Nowadays, visual sensing is one of the best choices for integrating sensor-based collision avoidance concepts in motion control system. Thanks to the possibility of monitoring large work spaces, the Microsoft Kinect [1] in combination with a set of useful tools (e.g., OpenCV [3], OpenNI [4], and PrimeSense [5] software libraries), many demands and requirements can already be met by a very cheap and powerful sensor system.

The classical way to use depth data is to project it into a robot oriented space, reassemble representations of obstacles in this space, and finally compute the information needed for the collision avoidance. Examples of this approach are [6], where Cartesian space control is used, and [9], in which obstacles are represented in the configuration space.

We propose to compute the information needed for the collision avoidance directly in the depth space (i.e., the image plane of the depth sensor), because only this smaller set of information is projected in a robot oriented space.

II. Main Idea

The core idea of our approach is derived from observations of human behaviors. When we (human beings) have to avoid obstacles, at least at the reflex level, we use visual feedback of rough estimation of relative distances between the obstacles and ourselves. The core of the idea is twofold: (1) *visual feedback*, we work directly in the visual space (i.e., depth space due to human stereo vision); (2) *rough estimation*, an exact measure of the relative distance is not needed. The Cartesian space projection is directly derived, and no a priori knowledge is used. For example, when we move in the dark we try to figure out where the obstacles are by projecting our remembrance of the environment into the 3D space.

A. Practical Implication

Beyond this human behavior imitation, many practical benefits in the field of robotics arise by using the depth space instead of other spaces. For example, if the Cartesian space is used, the steps to be accomplished are: project tens of thousand points into the Cartesian space; take into account the projection ray for side points in order to consider occluded points too; fit some primitive shapes to cover the surface of the obstacle; compute distances between the obstacles and the robot. If, instead, the depth space is used the steps are: project robot points in the depth space; compute distances between the obstacles and the robot; optionally project the results in the Cartesian space. The first difference is that by using primitive shapes (e.g., spheres), less than one hundred points are sufficient to cover the entire robot; therefore, the projection phase is quite faster, moreover the projected points describing the robot are perfectly known (e.g., by interoceptive position sensors), instead sensed points are very noisy and the projection could amplify the error. The distance between one robot point and one obstacle pixel can be evaluated with very simple formulas directly considering the occluded point. Since we are interested in a relative distance, projections of the distance into the Cartesian space are usually not needed. Once the distances are obtained, they are used to compute repulsive velocities or forces needed to avoid collisions.

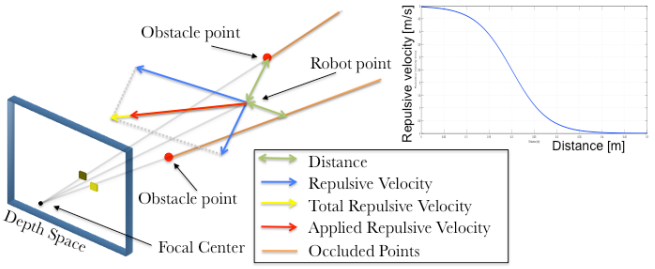

Fig. 1. Schematic example of the repulsive velocity computation. In this example, one robot point and two obstacle points are considered. The two possible distance computations are shown. The top right plot is the repulsive velocity as a function of the distance.

III. Repulsive Velocities

To obtain the repulsive velocities, the first step is to project the point of interest of the robot (center of the spheres which cover the robot) into the depth space, while the robot image is removed from the sensed data; otherwise the robot would be considered as an obstacle, too. For each point of interest of the robot, the distance to all obstacle points within a circle are computed. As shown in Fig. 1 there are only two possibilities: (1) if a robot point has a smaller depth than the obstacle, a simple Euclidean distance between two point is computed. Otherwise, (2) the occluded points have to be taken into account, and in this case, the depth of the obstacle is considered equal to the depth of the robot point. This is not the actual shorter distance, but following the guideline proposed in Sec. II, we do not waste computational time to add inessential information.

Each distance is associated with a repulsive velocity vector, whose magnitude is given by the function of the distance shown in Fig. 1, and whose direction connects the robot and the obstacle point. The total repulsive velocity is obtained by adding all repulsive velocity vectors that act in a robot point.

The actual repulsive velocity applied to the robot point has the same direction of the total repulsive velocity but the magnitude is related only to the minimum distance, neither the number of obstacle points nor the distance between near and far obstacle points has any influence on the magnitude of the repulsive velocity.

It should be noted that repulsive velocities of obstacle pixels are independent of each other, such that they can be computed concurrently, and, since all points within one circle are considered, we are even able to avoid also multiple objects.

IV. Obstacle Velocity

When the obstacle is non-static, moving the robot with a repulsive velocity that only takes into account the relative position between the obstacle and the robot is not sufficient. If the obstacle moves trough the robot with a velocity higher than the maximum robot velocity the collision cannot be avoided. The solution to this problem can be obtained from the observation of human behaviors as well. If the obstacle is fast, we move toward a direction normal to the velocity vector of the obstacle in order to move away from the obstacle trajectory as fast as possible.

The same concept is used in our approach: the obstacle velocity is estimated by considering the time variation of the repulsive velocity, then the orientation of the actual repulsive force is modified according to the estimated velocity. The faster an obstacle is, the more significant the change of orientation will be, and the orientation will be exactly orthogonal to the velocity vector if the obstacle velocity reaches the maximum velocity of the robot.

V. Collision Avoidance

Once the repulsive velocity has been computed, the manipulator executes it. Since the robot has to work close to the human, also the consideration of safety is very important. Discontinues and jerky motions make the human feel unsafe and uncomfortable, even though collisions are avoided. To obtain a smooth, jerk-limited, and executable trajectory, we use the Reflexxes Motion Libraries [2].

At present, we consider only the robot end effector to test our new collision avoidance concept, therefore we have to prevent collisions between unmodeled robot points and fixed obstacles (e.g., the table, on which the robot is mounted). We perform null space motion control to impede collisions between the robot elbow and the table in order to achieve this goal.

VI. Experimental Setup and Experiments

Our experiments have been performed with a KUKA Light-Weight-Robot IV, and the work space has been monitored by a Microsoft Kinect depth sensor [1] positioned at a planar distance of two meters and at a height of 1.2 meters w.r.t. the robot base frame. The implementation of this new collision avoidance approach is executed on an eight-core-CPU: four processors to execute the repulsive velocity computation, and the other four cores to enable visualization and robot motion control.

Fig. 2. Multiple overlapping image frames of the attached video showing the performance of the new collision avoidance strategy.

In this experiment, the task for the robot is to maintain a fixed configuration, while the human tries to touch the end effector. A multiple image frame overlapping of a experiment is shown in Fig. 2.

VII. Future Work

We are currently working on extending this new collision avoidance concept, such that the entire robot body can be taken into account; task prioritization must be used in order to take advantage of the redundancy of the robot. In the near future, whole-body tracking of the PrimeSense library [5] will be used to permit *desired contact* between the robot and the human (collaboration) while other parts of the human body and external obstacle are avoided in any case.

References

[1] Microsoft Kinect for Xbox 360. URL www.xbox.com/kinect.

[2] Reflexxes Motion Libraries. URL http://www.reflexxes.net.

[3] OpenCV (open source computer vision), . URL http://opencv.willowgarage.com/wiki/.

[4] OpenNI, . URL http://www.openni.org/.

[5] PrimeSense. URL http://www.primesense.com/.

[6] L. Bascetta, G. Magnani, P. Rocco, R. Migliorini, and M. Pelagatti. Anti-collision systems for robotic applications based on laser Time-of-Flight sensors. In *Advanced Intelligent Mechatronics (AIM), 2010 IEEE/ASME International Conference on*, pages 278 –284, july 2010.

[7] A. Bicchi and G. Tonietti. [Fast and Soft Arm Tactics: Dealing with the Safety-Performance Trade-Off in Robot Arms Design and Control. *IEEE Robotics and Automation Mag.*, 11:22–33, 2004.

[8] A. Bicchi, M.A. Peshkin, and J.E. Colgate. Safety for physical human-robot interaction. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 1335–1348. Springer, 2008.

[9] R. Schiavi, F. Flacco, and A. Bicchi. Integration of active and passive compliance control for safe human-robot coexistence. In *Proc. 2009 IEEE Int. Conf. on Robotics and Automation*, pages 259–264, 2009.