

Contact Tracking in Configuration Space for Haptic Rendering Purposes

Israel Vázquez

Jan Rosell

Institute of Industrial and Control Engineering
Technical University of Catalonia, Barcelona, SPAIN
email: israel.vazquez@upc.edu jan.rosell@upc.edu

Abstract

This paper introduces an efficient contact tracking procedure for the haptic rendering of contacts produced between simple convex polyhedral objects. The approach followed is based on the task configuration space and uses spatial and temporal coherence in order to be able to comply to the hard temporal constraints of the haptic servo loops. The use of the configuration space facilitates the knowledge of the type of contacts that take place between objects, allowing a better haptic rendering when face-face or edge-face contacts occur.

1 Introduction

Humans can interact with virtual worlds through the sense of touch by using haptic devices. The reaction forces and torques fed back to the haptic device are computed from the interference between the object attached to the user-manipulated probe and the objects of the virtual environment. This process is known as haptic rendering and must be carried out in less than one millisecond in order to obtain a smooth and stable response [11].

Many algorithms for interference detection are based on the local search of the contacting topological elements, like Young et al. [6] that propose an incremental algorithm based on the implicit, local construction of the Minkowski sums and on an iterative optimization of the penetrating distance value, or the approach of Johnson and Willemssen [5] that is based on a global search for local minimum distances, that are later continuously updated by a local tracking algorithm.

When several contacts take place simultaneously, the reaction force and torque is usually computed in an approximate way by the interpolation or the sum of the forces computed at each contact point [3, 11]. Alternatively, other

approaches use approximate discretized representations of the objects, like Garroway [2] that represents the manipulated object as a set of points each one with an associated force vector dependant on its location with respect to the obstacle, or McNeely et al. [8] that also discretize the obstacles using an octree of regular small volumes called voxels that allow to compute the penetrating volume.

For the haptic rendering of virtual assembly tasks between simple polyhedral objects, face-face contacts and edge-face contacts are common and, moreover, compliant motions are usually performed maintaining these types of contacts. In these situations, the previous approaches do not provide a good enough haptic rendering. A better haptic rendering can then be achieved if the kind of contacts that take place between the manipulated object and the obstacles in the environment is taken into account, as proposed by You and Xiao [13]. These authors compute the reaction forces and torques once the contact situation has been identified as a collection of principal contacts¹. Contact identification is done by using classical collision detection algorithms, reasoning procedures and a precomputed graph of possible contact situations.

2 Haptic Rendering using \mathcal{C} -space

The task configuration space (\mathcal{C} -space) has been extensively used in motion planning problems [7]. The \mathcal{C} -space explicitly captures the contact constraints (represented as the border of the \mathcal{C} -space obstacles) and the motion freedom of the mobile object (that is mapped to a point). Therefore, the \mathcal{C} -space can be a useful tool to know the type of contacts that take place between the mobile object and the obstacles in the workspace, and to use this information for the computation of the reaction force and torque, as suggested in [13]. Figure 1 shows the proposal of a haptic rendering framework based on the task configuration

¹This work was partially supported by the CICYT projects DPI2002-03540 and DPI2004-03104. Israel Vázquez has a scholarship from COFAA-IPN and COTEPABE-SUPERA

¹A principal contact is a contact between a pair of topological elements (vertices, edges or faces) that are not the boundary of other topological elements in contact.

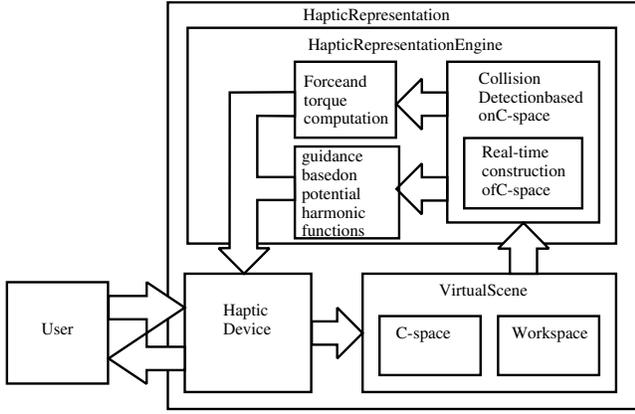


Figure 1: Haptic rendering using \mathcal{C} -space.

space, that was first explored in a preliminary version [12] providing promising results.

The use of \mathcal{C} -space also has other benefits. The visualization of \mathcal{C} -space, together with that of physical space, may aid the user in performing the constrained motions involved in low-clearance assembly tasks (e.g. [4] [9]). Also, it allows the use of gross-motion planning techniques to compute a force field to guide the user in performing the virtual assembly task.

The disadvantage of using the \mathcal{C} -space is the complexity of its computation. Nevertheless, for haptic rendering purposes, the \mathcal{C} -space only needs to be computed locally. This paper proposes an efficient procedure with that aim, based on the tracking of the current contacts taking place or of the nearest potential contacts (if no contact is taking place). Convex polyhedra and the existence of spacial and temporal coherence are assumed. The paper is structured as follows. Section 3 introduces the \mathcal{C} -space and its modelling, which is the base of the contact tracking procedure presented in Section 4. Section 5 sketches its use in the haptic rendering loop and, finally, Section 6 concludes the work.

3 Configuration Space

3.1 Definition

Let \mathcal{A} and \mathcal{B} be two polyhedra describing the manipulated object and a static object, respectively. Let \mathcal{F}_A be reference frame attached to \mathcal{A} and \mathcal{F}_W be the fixed reference frame of the workspace. Let $q^A = (x^A, \Theta^A)$ be a configuration of \mathcal{A} , where x^A and Θ^A describe, respectively, the position and the orientation of \mathcal{F}_A with respect to \mathcal{F}_W . Let also $q_0^A = (x_0^A, \Theta_0^A)$ be the current configuration of \mathcal{A} .

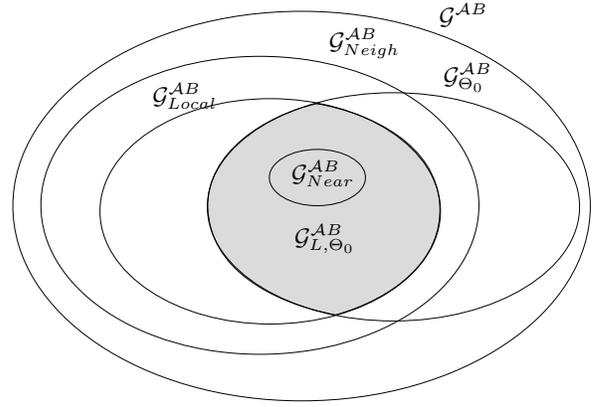


Figure 2: Subgraphs of the graph \mathcal{G}^{AB} that describes a \mathcal{C} -obstacle \mathcal{CO}_B .

The Configuration Space (\mathcal{C} -space) of \mathcal{A} is the space defined by all of its configurations. The subset of configurations where there is interference between \mathcal{A} and \mathcal{B} is called the \mathcal{C} -obstacle \mathcal{CO}_B . The border of \mathcal{CO}_B is composed of \mathcal{C} -faces, each \mathcal{C} -face being the subset of configurations of a five-dimensional hyper-surface $f(q^A) = 0$ where a given basic contact takes place. There are three types of basic contacts between \mathcal{A} and \mathcal{B} :

- Type-A: a face of \mathcal{A} against a vertex of \mathcal{B} .
- Type-B: a vertex of \mathcal{A} against a face of \mathcal{B} .
- Type-C: an edge of \mathcal{A} against an edge of \mathcal{B} .

For each type of basic contact, an applicability condition can be defined to determine if for a given orientation of \mathcal{A} the contact can take place [1].

For orientation Θ_0^A , the subset of the \mathcal{C} -faces corresponding to the basic contacts that satisfy their applicability condition are planar polygons over the corresponding planes $f(x^A, \Theta_0^A) = 0$. These are the faces of the 3D polyhedron that represent the \mathcal{C} -obstacle for that orientation Θ_0^A .

3.2 Modelling of \mathcal{C} -obstacles

From now on let consider all polyhedra convex (i.e. if polyhedra are non-convex they are previously decomposed into convex ones). Then, the modelling of the (convex) \mathcal{C} -obstacles is done as follows.

Each \mathcal{C} -obstacle \mathcal{CO}_B will be represented as a graph, \mathcal{G}^{AB} , that captures its topology. The nodes of \mathcal{G}^{AB} are all the possible basic contacts between the topological elements of \mathcal{A} and those of \mathcal{B} , and its arcs show the neighboring relationship, as computed in the next subsection. The neighbor nodes of a node of \mathcal{G}^{AB} are called \mathcal{G} -neighbors.

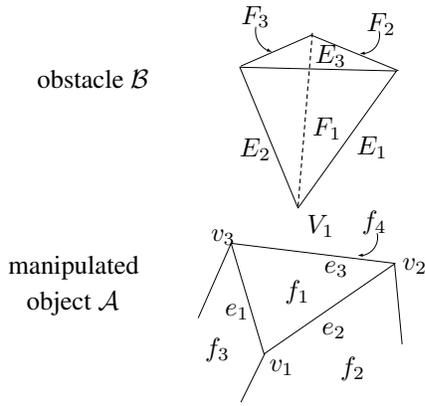


Figure 3: Type-A basic contact (f_1, V_1) . Some of the 24 basic contacts that neighbor on (f_1, V_1) are (f_2, V_1) , (e_1, E_1) and (v_3, F_1) .

The following subgraphs of \mathcal{G}^{AB} are defined (Figure 2):

- The subset of nodes of \mathcal{G}^{AB} corresponding to the set of \mathcal{C} -faces that are nearest to x_0^A form a subgraph called *near subgraph*, \mathcal{G}_{Near}^{AB} (distance computation is tackled in Section 3.4). Note that, since \mathcal{A} and \mathcal{B} are convex, \mathcal{G}_{Near}^{AB} will contain only one node except for the orientations where face-face contacts or edge-face contacts are possible.
- The nodes of \mathcal{G}_{Near}^{AB} and their \mathcal{G} -neighbors form a subgraph called *neighbor subgraph*, \mathcal{G}_{Neigh}^{AB} .
- The nodes of \mathcal{G}_{Neigh}^{AB} whose distance to x_0^A is close to the distance from \mathcal{G}_{Near}^{AB} to x_0^A form a subgraph called *local subgraph*, \mathcal{G}_{Local}^{AB} .
- The subset of nodes of \mathcal{G}^{AB} that can take place for a given orientation Θ_0^A , i.e. those satisfying the applicability condition, form a subgraph called *applicability subgraph*, $\mathcal{G}_{\Theta_0}^{AB}$.
- The subset of nodes that belong to $\mathcal{G}_{Local}^{AB} \cap \mathcal{G}_{\Theta_0}^{AB}$ form a subgraph called the *local applicability subgraph*, $\mathcal{G}_{L, \Theta_0}^{AB}$.

Due to the spatial and temporal coherence, i.e. due to the fact that the \mathcal{C} -space changes very slightly around q_0^A for a motion of \mathcal{A} between two consecutive instants of time, only the nodes of $\mathcal{G}_{L, \Theta_0}^{AB}$ are to be considered for hatpic rendering purposes. When the orientation of \mathcal{A} changes, $\mathcal{G}_{\Theta_0}^{AB}$ may change (and thus $\mathcal{G}_{L, \Theta_0}^{AB}$) since some basic contacts may no longer be possible (i.e. their applicability condition may no longer hold) and others may become possible. Moreover, if the position of \mathcal{A} changes, $\mathcal{G}_{L, \Theta_0}^{AB}$ may change since the nearest basic contact(s) rep-

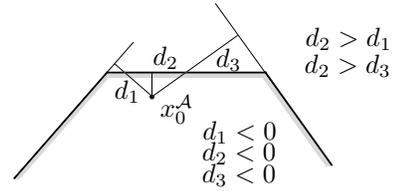


Figure 4: Distances from x_0^A to the supporting planes of the \mathcal{C} -faces. All distances are negative and therefore a collision exists, contact 2 is taking place since $d_2 > d_1$ and $d_2 > d_3$.

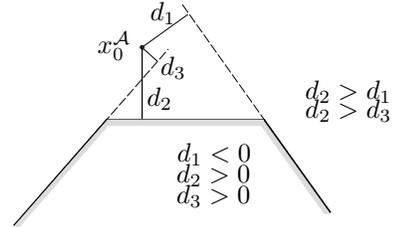


Figure 5: Distances from x_0^A to the supporting planes of the \mathcal{C} -faces. Not all distances are negative and therefore a collision does not exist, contact 2 is the nearest contact since $d_2 > d_1$ and $d_2 > d_3$.

resented by \mathcal{G}_{Local}^{AB} may change. The update of $\mathcal{G}_{L, \Theta_0}^{AB}$ is tackled in Section 4.

3.3 Neighborhood

The neighboring relationship of the nodes of the graph \mathcal{G}^{AB} that captures the topology of a \mathcal{C} -obstacle is obtained as follows. Let F , E and V represent, respectively, a face, an edge and a vertex of a polyhedron, and let the following neighborhood operators be defined as:

- $N_v(F)$: gives the vertices of face F .
- $N_e(F)$: gives the edges of face F .
- $N_f(F)$: gives the faces that contain an edge of $N_e(F)$, excluding F .
- $N_f(V)$: gives the faces that contain vertex V .
- $N_e(V)$: gives the edges that contain vertex V .
- $N_v(V)$: gives the vertices of the edges of $N_e(V)$, excluding V .
- $N_e(E)$: gives the edges that share a vertex with E .

Then, the procedure to compute the arcs of \mathcal{G}^{AB} has the following steps:

1. Connect each type-A basic contact (F_A, V_B) with type-A basic contacts $(N_f(F_A), V_B)$ and $(F_A, N_v(V_B))$, with type-B basic contacts $(N_v(F_A), N_f(V_B))$, and with type-C basic con-

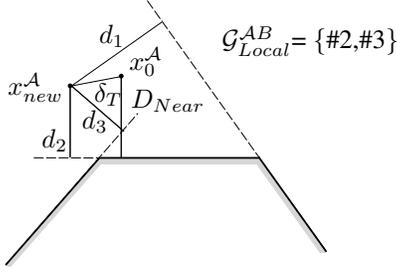


Figure 6: After a translation from x_0^A to x_{new}^A , the local subgraph \mathcal{G}_{Local}^{AB} is composed of the nodes corresponding to \mathcal{C} -faces #2 and #3 since for those contacts $|D_{Near} - d_2| < (\delta_T + \delta_R)$ and $|D_{Near} - d_3| < (\delta_T + \delta_R)$.

tacts $(N_e(F_A), N_e(V_B))$.

2. Connect each type-B basic contact (V_A, F_B) with type-B basic contacts $(N_v(V_A), F_B)$ and $(V_A, N_f(F_B))$, and with type-C basic contacts $(N_e(V_A), N_e(F_B))$.
3. Connect each type-C basic contact (E_A, E_B) with type-C basic contacts $(E_A, N_e(E_B))$ and $(N_e(E_A), E_B)$.

As an example Figure 3 shows some of the \mathcal{G} -neighbors of a type-A basic contact.

3.4 Determination of distances

The distance from the current position x_0^A to the supporting plane $f_i(x^A, \Theta_0^A) = 0$ of the \mathcal{C} -face of contact i for the orientation Θ_0^A is computed as $d_i = f_i(x_0^A, \Theta_0^A)$. If all the distances are negative then x_0^A is inside \mathcal{CO}_B and a collision between \mathcal{A} and \mathcal{B} is taking place in physical space.

Since the \mathcal{C} -obstacles are convex and the distance is measured from x_0^A to the planes containing the \mathcal{C} -faces, then it results that the \mathcal{C} -face that is nearest to x_0^A is the one that has the greater value of the distance d_i , as illustrated in Figures 4 and 5. If a collision exist, the actual contact that is taking place is the one that has the greater distance.

4 Contact Tracking / Collision detection

4.1 Procedure

Let D_{Near} be the distance from the current position x_0^A to the nearest \mathcal{C} -face, and let $q_{new}^A = (x_{new}^A, \Theta_{new}^A)$ be the new configuration of the manipulated object fixed by the user as he manipulates the probe of the haptic device.

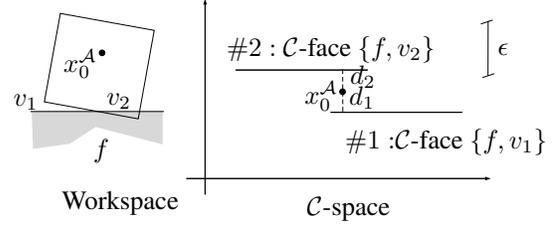


Figure 7: Although $d_1 > d_2$, D_{Near} is set to d_2 because contact #1 satisfies the applicability condition only because some tolerance is permitted. Since $|D_{Near} - d_1| < \epsilon$ both contacts belong to \mathcal{G}_{Near}^{AB} , i.e. it is considered a contact situation involving the two basic contacts.

Each time there is a change in the position and/or orientation of \mathcal{A} , i.e. $q_{new}^A \neq q_0^A$, the following steps are executed for each \mathcal{CO}_B in order to recompute the corresponding subgraph \mathcal{G}_{Near}^{AB} , and verify if a collision exists:

1. Compute the distances d_i from the new position x_{new}^A to all the nodes of \mathcal{G}_{Neigh}^{AB} , as explained in subsection 3.4.
2. Set \mathcal{G}_{Local}^{AB} with all the nodes of \mathcal{G}_{Neigh}^{AB} satisfying that:

$$|D_{Near} - d_i| < (\delta_T + \delta_R) \quad (1)$$

where $\delta_T = |x_{new}^A - x_0^A|$ and δ_R is a fixed small value. This step is performed to prune all the nodes of \mathcal{G}_{Neigh}^{AB} whose \mathcal{C} -faces are, due to the spatial and temporal coherence assumed, too far from x_0^A to become the new nearest \mathcal{C} -face (Figure 6). The values δ_T and δ_R are fixed to consider the translations and the rotations, respectively.

3. Set $\mathcal{G}_{L, \Theta_0}^{AB}$ with all the nodes of \mathcal{G}_{Local}^{AB} that satisfy the applicability condition (some tolerance is allowed, i.e. those nearly satisfying the condition are also accepted).
4. Order the nodes of $\mathcal{G}_{L, \Theta_0}^{AB}$ in descending values of d_i . Set D_{Near} with the maximum d_i such that the corresponding basic contact satisfies the applicability condition with null tolerance (Figure 7).
5. Set \mathcal{G}_{Near}^{AB} with the node corresponding to D_{Near} and all the nodes that satisfy:

$$|D_{Near} - d_j| < \epsilon \quad (2)$$

ϵ is a fixed small value that allows some tolerance in the handling of face-face and edge-face contact situations (Figure 7).

6. A collision is taking place if $D_{near} \leq 0$.

As an example Figure 10 shows four snapshots of a motion of the manipulated object both in physical space and

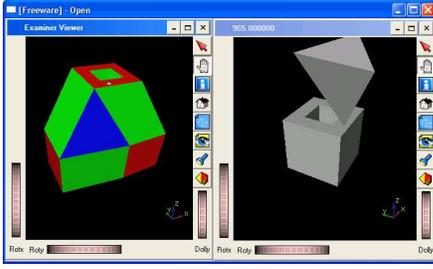


Figure 8: Interaction between a convex object and a concave one. Left: \mathcal{C} -space, Right: physical workspace.

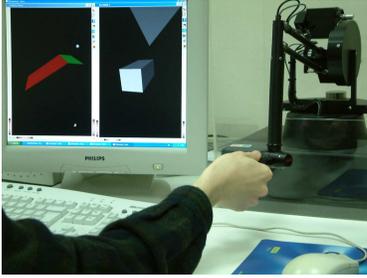


Figure 9: Haptic interaction with the virtual assembly task using the 6 d.o.f. Phantom haptic device.

\mathcal{C} -space, and a table with the nodes of the corresponding graphs. It can be seen how the nearest \mathcal{C} -face and the local graph $\mathcal{G}_{L, \Theta_0}^{AB}$ change. The graph \mathcal{G}^{AB} of this simple assembly task is composed of 128 nodes (\mathcal{A} is composed of 4 faces, 4 vertices and 6 edges and \mathcal{B} is composed of 6 faces, 8 vertices and 12 edges).

4.2 Implementation Issues

The developed procedures have been implemented in C++ on a PC computer at 1.4 GHz. Contact tracking is implemented on a separate execution thread. The tracking of contact situations involving up to four basic contacts can be achieved in less than 0.5 ms. The algorithm that implements the visualization loop runs on an independent execution thread and at a lower rate. The interface has been developed using Qt and OpenInventor. Several simple assembly tasks have been considered. Figure 8 shows a peg-into-hole assembly task (the hole has been previously decomposed into five convex polyhedra). A 6 d.o.f. Phantom haptic device has been used for the experiments (Figure 9).

5 Haptic rendering

If a contact occurs, i.e. when $D_{Near} \leq 0$, then the reaction force and torque has to be computed and fed back to

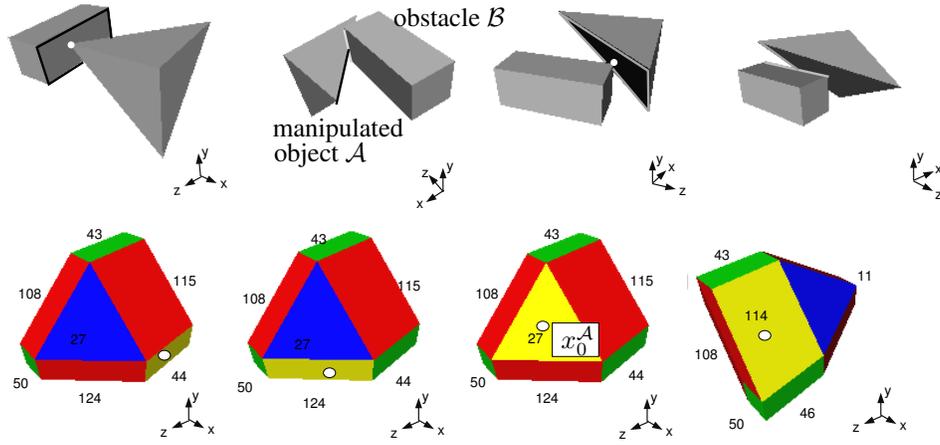
the haptic device. If \mathcal{G}_{Near}^{AB} is composed of only one node, a single basic contact is taking place. Then, the reaction force is in the direction of the normal to the corresponding \mathcal{C} -face, and the torque is computed from the knowledge of the contact point. If \mathcal{G}_{Near}^{AB} is composed of more than one node, a face-face or an edge-face are occurring. In these cases all the \mathcal{C} -faces are coplanar and hence the reaction force is in the direction normal to them, and the reaction torque must be computed taking into account the basic contacts involved [10].

6 Conclusions

When virtual assembly tasks are executed using a haptic device, the haptic rendering procedures must properly compute the reaction forces and torques arising at a face-face or at an edge-face contact, since compliant motions maintaining those contacts are usually performed. In order to obtain a smooth haptic rendering in these situations, the information of the current type of contact taking place is necessary. Taking into account this need, this paper has introduced an approach based on the \mathcal{C} -space. To avoid the computational complexity of the \mathcal{C} -space construction, a procedure to compute it locally has been proposed based on the tracking of the current contacts taking place or of the nearest potential contacts (if no contact is taking place). The topology of a \mathcal{C} -obstacle is represented as a graph. Taking into account spatial and temporal coherence, the neighborhood of the nearest contact is maintained as a subgraph, that is updated each time a motion is performed. This allows an efficient contact tracking necessary for the haptic rendering.

References

- [1] B. R. Donald. On motion planning with six degrees of freedom: Solving the intersection problems in configuration space. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 536–541, 1985.
- [2] D. Garraway. 6dof haptic rendering using geometric algebra. In *Proc. of the Int Workshop on Haptic Virtual Environments and Their Applications*, pages 103–107, 2002.
- [3] A. Gregory, A. Mascarenhas, S. Ehmann, M. Lin, and D. Manocha. Six degree-of-freedom haptic display of polygonal models. In *Proc. on Visualization*, pages 139–146, 2000.
- [4] I. Ivanisevic and V. Lumelsky. Augmenting human performance in motion planning tasks - the configuration space approach. In *Proc. of the IEEE Int. Conf. on Robotics and Autom.*, pages 2649–2654, 2001.



		\mathcal{G}_{Near}^{AB}					
		\mathcal{G}_{Local}^{AB}					
		$\mathcal{G}_{L, \Theta_0}^{AB}$					
		\mathcal{G}_{Near}^{AB}					
S1	n^o	44	124 115	8 11 23 24 40 110 112 67	4 6 7 10 20 22 26 27 32 45 46 47 52	48 116 118 121 122 127 62 64	
	dist	0.42	-3.53 -2.25	4.08 3.32 -0.14 3.40 -3.65 -3.65 -3.63 -3.34	...		
S2	n^o	124	20 27	23 64 70	24 44 46 48 50 106 112 122 123 126 127		
	dist	0.17	-3.11 -2.77	2.00 -3.12 -3.11	...		
S3	n^o	27	115	46	11 23 24 25 26 31 40 42 43 44 47 48 50 51 106	108 109 112 114 124 126 127	
	dist	-0.19	-3.80	3.41	...		
S4	n^o	114	–	27	9 11 25 42 43 46 47 102 108 111	112 113 115 126 120	
	dist	1.16	–	3.16	...		

Figure 10: Snapshots of a motion (two translations followed by a rotation) of the manipulated object both in physical space and C -space. The table shows the nodes of \mathcal{G}_{Near}^{AB} , \mathcal{G}_{Local}^{AB} and $\mathcal{G}_{L, \Theta_0}^{AB}$ corresponding to each snapshots.

- [5] D. E. Johnson and P. Willemsen. Accelerated haptic rendering of polygonal models through local descent. In *Proc. of the 12th Int. Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 18 – 23, 2004.
- [6] Y. J. Kim, M. C. Lin, and D. Manocha. Incremental penetration depth estimation between convex polytopes using dual-space expansion. *IEEE Trans. on Visualization and Computer Graphics*, 10(2):152 – 163, 2004.
- [7] T. Lozano-Pérez. Spatial planning: A configuration space approach. *IEEE Trans. Comput.*, 32(2):108–120, 1983.
- [8] W. McNeely, K. Puterbaugh, and J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. In *Proc. of ACM SIGGRAPH*, pages 401–408, 1999.
- [9] K. Morishige and H. Noborio. A tele-operation support system with the help of dual views of cartesian and configuration space. In *Proc. of the IEEE Int. Workshop on Robot and Human Interactive Communication*, pages 382–387, 2000.
- [10] J. Rosell and I. Vázquez. Haptic rendering of compliant motions using contact tracking in cspace. Accepted to the IEEE Int. Conf. on Robotics and Automation 2005.
- [11] D. C. Ruspini, K. Kolarov, and O. Khatib. The haptic display of complex graphical environments. In *Proc. of the 24th annual Conf. on Computer graphics and Interactive Techniques*, pages 345–352, 1997.
- [12] I. Vázquez and J. Rosell. Using configuration space for the haptic rendering of assembly tasks. In *Proc. of the World Automation Congress*, 2004.
- [13] S. You and J. Xiao. Haptic display of contact states. In *Proc. of the 4th. World Congress on Intelligent Control and Automation*, pages 2854–2859, 2002.