

Fill and Transfer: A Simple Physics-based Approach for Containability Reasoning Supplementary Materials

1. User Study Details

We provide details of our user study. Figures 1 to 5 display all the objects in our user study, together with the user annotations and outputs of our experiments. Tables 2 and 3 further list the user annotation statistics and the results of our experiments.

All objects are randomly rotated before being displayed to the user to annotate. For each object, if more than 30% of the users vote that it is a container, we conclude that it is a possible container. In addition, if more than 80% of the users vote that all its tilt axes (transfer directions) are equally-likely, we conclude that all its tilt axes (transfer directions) are equally-likely.

As noted in our paper, our algorithm concludes that the object is not a container if the percentage volume of its containee out of the total volume of the container plus containee is less than 10%, assuming the object has been filled up along its best filling axis. In determining whether all tilt axes (transfer directions) are equally-likely, our algorithm computes the normalized standard deviation of the Z_g (*i.e.* N sum) obtained by tilting the object about different tilt axes. Our algorithm determines that all tilt axes (transfer directions) are equally-likely if the normalized standard deviation is less than 0.1.

We compare our proposed tilt axis with a user-annotated tilt axis as follows. If our proposed tilt axis deviates from a user-annotated tilt axis by less than an angular error tolerance, we count it as a match. Table 1 lists the precision and recall versus angular error tolerance in the tilt axis estimation.

We list all the objects used, in four categories, as follows:

True Positives. These are the objects that most users vote as containers and our algorithm also outputs as containers. Figures 1, 2 and 3 show the objects. For each object, we also display whether the user study concludes and our algorithm outputs that all the tilt axes (transfer directions) are equally-likely.

There is an interesting observation. Our algorithm outputs that objects 3, 20 and 22 have equally-likely tilt axes (transfer directions), however, about half of the users do not

think so (see Table 2). We note the presence of a handle or a grasping region on these objects, and from the annotations we observe that a certain amount of users choose to tilt the objects as if a hand was grasping the handle or grasping region. This suggests that such regions should serve as meaningful features in determining the transfer directions.

True Negatives. These are the objects that most users vote as non-containers and our algorithm also outputs as non-containers. Figure 4 shows the objects.

False Positives. These are the objects that most users vote as non-containers and our algorithm outputs as containers. Figure 5 shows the objects. Object 69 is flat and hence is unstable in holding liquid. This might be the reason why users in general do not think it is a container. For object 70 (a jar), users may not notice that there is a hole at the mouth that water can flow through.

False Negatives. These are the objects that most users vote as containers and our algorithm outputs as non-containers. Figure 6 shows the objects. As the contained volume is small, our simple heuristics of counting the percentage volume of the containee out of the total volume of the container plus containee fails to identify these objects as potential containers.

Tolerance	10°	20°	30°	40°	50°	60°	70°	80°	90°
Precision	0.75	0.82	0.88	0.91	0.95	0.97	0.97	0.97	0.97
Recall	0.24	0.43	0.55	0.65	0.75	0.81	0.83	0.85	0.88

Table 1. Precision and recall versus angular error tolerance in tilt axis estimation.

ID	Input	Filling Axis	Tilt Axis	ID	Input	Filling Axis	Tilt Axis
1				2			
3				4			
5				6			
7				8			
9				10			
11				12			
13				14			

Figure 1. **True Positives.** Objects that most users vote as containers and our algorithm also outputs as containers. In the *Filling Axis* column, each green dashed line corresponds to a user-annotated filling axis. The red line corresponds to the average of the user-annotated filling axes, and the blue line corresponds to the filling axis estimated by our algorithm. In the *Tilt Axis* column, each green dashed line corresponds to a user-annotated tilt axis, and the blue lines correspond to the tilt axes estimated by our algorithm. *UE* means our user study concludes that all tilt axes (transfer directions) are equally-likely. *E* means our algorithm outputs that all tilt axes (transfer directions) are equally-likely.

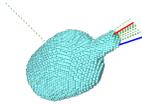
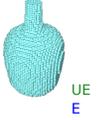
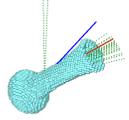
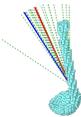
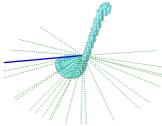
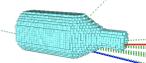
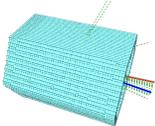
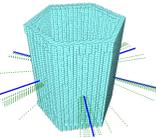
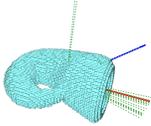
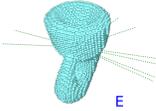
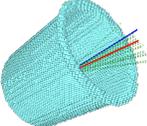
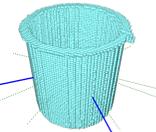
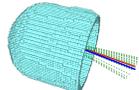
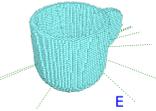
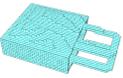
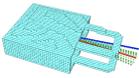
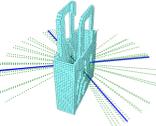
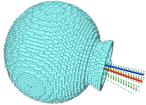
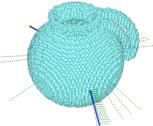
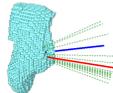
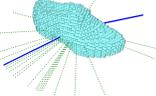
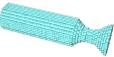
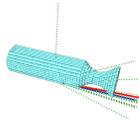
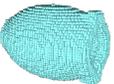
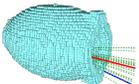
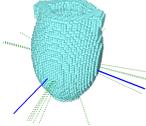
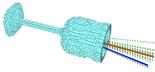
ID	Input	Filling Axis	Tilt Axis	ID	Input	Filling Axis	Tilt Axis
15			 UE E	16			 UE E
17				18			 UE E
19				20			 E
21				22			 E
23				24			
25				26			 UE E
27				28			 UE E

Figure 2. **True Positives (continued)**. Refer also to Figure 1 for description.

ID	Input	Filling Axis	Tilt Axis	ID	Input	Filling Axis	Tilt Axis
29				30			
31				32			
33				34			
35				36			
37				38			
39				40			

Figure 3. **True Positives (continued)**. Refer also to Figure 1 for description.

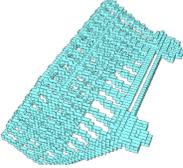
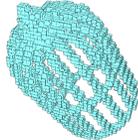
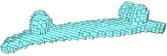
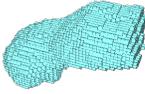
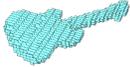
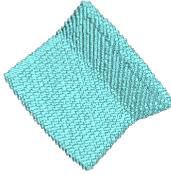
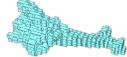
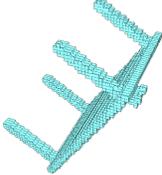
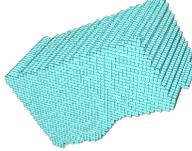
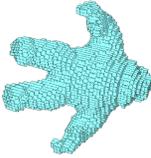
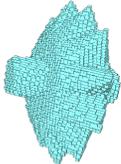
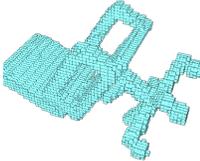
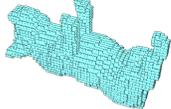
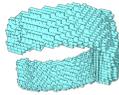
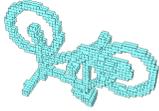
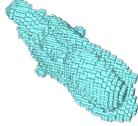
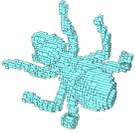
ID	Input	ID	Input	ID	Input	ID	Input
41		42		43		44	
45		46		47		48	
49		50		51		52	
53		54		55		56	
57		58		59		60	
61		62		63		64	
65		66		67		68	

Figure 4. **True Negatives.** Objects that most users vote as non-containers and our algorithm also outputs as non-containers.

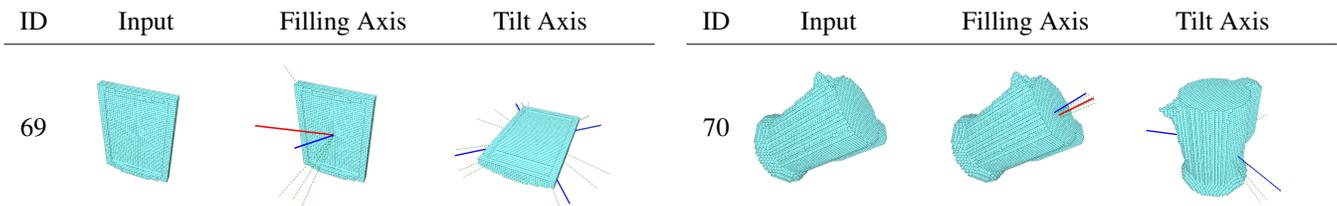


Figure 5. **False Positives.** Objects that most users vote as non-containers and our algorithm outputs as containers. Refer also to Figure 1 for description.

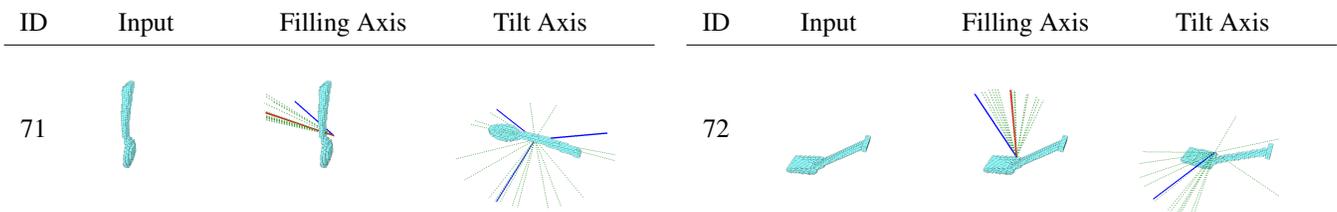


Figure 6. **False Negatives.** Objects that most users vote as containers and our algorithm outputs as non-containers. For illustration purposes, we still run our algorithm to estimate the filling axis and tilt axes even though our algorithm has concluded that the objects are non-containers. Refer also to Figure 1 for description.

ID	Users		Ours		Users		Ours		Best filling axis error
	container vote %	container	containeer volume %	container	equally-likely tilt vote %	equally-likely tilt	normalized SD	equally-likely tilt	
1	100.00%	yes	85.01%	yes	100.00%	yes	0.06	yes	18.93 °
2	91.67%	yes	56.33%	yes	68.18%	no	0.16	no	5.04 °
3	95.83%	yes	12.33%	yes	52.17%	no	0.04	yes	19.06 °
4	100.00%	yes	31.24%	yes	100.00%	yes	0.11	yes	19.38 °
5	100.00%	yes	74.40%	yes	91.67%	yes	0.05	yes	3.93 °
6	100.00%	yes	84.92%	yes	91.67%	yes	0.07	yes	15.93 °
7	100.00%	yes	45.19%	yes	0.00%	no	0.32	no	29.46 °
8	62.50%	yes	51.01%	yes	6.67%	no	0.21	no	31.06 °
9	100.00%	yes	67.94%	yes	4.17%	no	0.21	no	18.90 °
10	95.83%	yes	69.25%	yes	0.00%	no	0.17	no	16.22 °
11	95.83%	yes	83.27%	yes	8.70%	no	0.58	no	31.76 °
12	87.50%	yes	44.86%	yes	9.52%	no	0.20	no	3.53 °
13	100.00%	yes	72.92%	yes	4.17%	no	0.62	no	5.69 °
14	91.67%	yes	17.30%	yes	45.45%	no	0.17	no	30.84 °
15	75.00%	yes	73.34%	yes	94.44%	yes	0.03	yes	9.42 °
16	100.00%	yes	46.36%	yes	95.83%	yes	0.03	yes	13.33 °
17	100.00%	yes	40.75%	yes	37.50%	no	0.13	no	8.78 °
18	95.83%	yes	56.16%	yes	91.30%	yes	0.02	yes	28.45 °
19	100.00%	yes	86.31%	yes	25.00%	no	0.30	no	7.23 °
20	100.00%	yes	69.87%	yes	70.83%	no	0.04	yes	36.36 °
21	100.00%	yes	79.31%	yes	8.33%	no	0.22	no	10.28 °
22	100.00%	yes	72.92%	yes	59.09%	no	0.11	yes	8.13 °
23	90.91%	yes	70.15%	yes	5.00%	no	0.26	no	7.67 °
24	100.00%	yes	81.40%	yes	45.45%	no	0.29	no	6.40 °
25	90.91%	yes	65.83%	yes	5.00%	no	0.26	no	32.40 °
26	95.45%	yes	56.37%	yes	80.95%	yes	0.03	yes	15.07 °
27	100.00%	yes	57.90%	yes	9.09%	no	0.15	no	29.46 °
28	100.00%	yes	54.00%	yes	95.45%	yes	0.02	yes	12.92 °
29	100.00%	yes	78.27%	yes	95.45%	yes	0.08	yes	10.57 °
30	100.00%	yes	73.87%	yes	77.27%	no	0.07	yes	3.04 °
31	100.00%	yes	71.71%	yes	95.45%	yes	0.05	yes	4.05 °
32	100.00%	yes	46.47%	yes	59.09%	no	0.06	yes	4.46 °
33	90.91%	yes	73.18%	yes	95.00%	yes	0.12	yes	10.41 °
34	100.00%	yes	59.43%	yes	100.00%	yes	0.07	yes	5.01 °
35	100.00%	yes	77.44%	yes	22.73%	no	0.11	yes	15.16 °
36	100.00%	yes	80.24%	yes	95.45%	yes	0.09	yes	4.27 °
37	100.00%	yes	49.73%	yes	13.64%	no	0.45	no	31.87 °
38	95.45%	yes	75.41%	yes	0.00%	no	0.32	no	29.88 °
39	72.73%	yes	58.65%	yes	43.75%	no	0.21	no	15.35 °
40	77.27%	yes	74.81%	yes	17.65%	no	0.31	no	23.87 °

Table 2. **User annotations and results of our experiments.** Each row corresponds to one object indexed by its ID. *Users-container vote %* refers to the percentage of users who vote this object as a container. *Users-container* indicates whether our user study concludes that this object is a container. *Ours-containeer volume %* refers to the percentage volume of the containee out of the total volume of the container plus containee, after the object has been filled up along the best filling axis. *Ours-container* indicates whether our algorithm outputs this object as a container. *Users-equally-likely tilt vote %* refers to the percentage of users who vote that all tilt axes are equally-likely. *Users-equally-likely tilt* indicates whether our user study concludes that all tilt axes are equally-likely. *Ours-normalized SD* refers to the normalized standard deviation of the Z_g (i.e. N sum) obtained by tilting the object about different tilt axes. *Users-equally-likely tilt* indicates whether our algorithm outputs that all tilt axes are equally-likely. *Best filling axis error* refers to the angular error between our estimated best filling axis and the average of the user-annotated filling axes.

ID	Users		Ours		Users		Ours		Best filling axis error
	container vote %	container	containeer volume %	container	equally-likely tilt vote %	equally-likely tilt	normalized SD	equally-likely tilt	
41	0.00%	no	0.24%	no	-	-	-	-	-
42	0.00%	no	0.08%	no	-	-	-	-	-
43	0.00%	no	0.14%	no	-	-	-	-	-
44	0.00%	no	0.06%	no	-	-	-	-	-
45	0.00%	no	0.00%	no	-	-	-	-	-
46	29.17%	no	0.19%	no	-	-	-	-	-
47	0.00%	no	0.03%	no	-	-	-	-	-
48	0.00%	no	1.37%	no	-	-	-	-	-
49	0.00%	no	0.00%	no	-	-	-	-	-
50	4.17%	no	0.36%	no	-	-	-	-	-
51	12.50%	no	0.04%	no	-	-	-	-	-
52	0.00%	no	0.75%	no	-	-	-	-	-
53	0.00%	no	0.06%	no	-	-	-	-	-
54	0.00%	no	0.27%	no	-	-	-	-	-
55	4.55%	no	0.05%	no	-	-	-	-	-
56	0.00%	no	0.00%	no	-	-	-	-	-
57	0.00%	no	0.07%	no	-	-	-	-	-
58	0.00%	no	0.12%	no	-	-	-	-	-
59	0.00%	no	0.10%	no	-	-	-	-	-
60	0.00%	no	0.22%	no	-	-	-	-	-
61	0.00%	no	0.02%	no	-	-	-	-	-
62	4.55%	no	4.62%	no	-	-	-	-	-
63	0.00%	no	0.04%	no	-	-	-	-	-
64	0.00%	no	7.88%	no	-	-	-	-	-
65	0.00%	no	0.04%	no	-	-	-	-	-
66	0.00%	no	0.72%	no	-	-	-	-	-
67	0.00%	no	7.25%	no	-	-	-	-	-
68	0.00%	no	0.22%	no	-	-	-	-	-
69	25.00%	no	29.10%	yes	16.67%	no	0.41	no	31.10 °
70	16.67%	no	75.16%	yes	0.00%	no	0.44	no	7.85 °
71	86.36%	yes	6.96%	no	5.26%	no	0.17	no	31.51 °
72	72.73%	yes	4.55%	no	6.25%	no	0.37	no	27.88 °

Table 3. User annotations and results of our experiments (continued). Refer also to Table 2 for description.

2. Mapping Sample Points to Unit Sphere

To prove that Q maps s_i back to the unit sphere, we look at the length of the projection \hat{s}_i ,

$$\begin{aligned}
 \hat{s}_i^T \hat{s}_i &= Q^T Q \\
 &= (\mathbf{f}^{(t)})^T \mathbf{f}^{(t)} \cos^2(\|\mathbf{u}_i\|) + \frac{\mathbf{u}_i^T \mathbf{u}_i}{\|\mathbf{u}_i\|^2} \sin^2(\|\mathbf{u}_i\|) \\
 &\quad + (\mathbf{f}^{(t)})^T \mathbf{u}_i \frac{2 \cos(\|\mathbf{u}_i\|) \sin(\|\mathbf{u}_i\|)}{\|\mathbf{u}_i\|} \\
 &= \cos^2(\|\mathbf{u}_i\|) + \sin^2(\|\mathbf{u}_i\|) + 0 \tag{1} \\
 &= 1 \tag{2}
 \end{aligned}$$

Equation (1) follows because $\mathbf{f}^{(t)}$ lies on the unit sphere, therefore $(\mathbf{f}^{(t)})^T \mathbf{f}^{(t)} = 1$; $\mathbf{u}_i^T \mathbf{u}_i = \|\mathbf{u}_i\|^2$; $(\mathbf{f}^{(t)})^T \mathbf{u}_i = 0$ as \mathbf{u}_i lies on the tangent plane about $\mathbf{f}^{(t)}$. It follows from Equation (2) that \hat{s}_i is a unit vector and hence lies on the unit sphere. For details of the Riemannian logarithm map, please refer to [1, 2].

3. Adding Noise

As a simple test for robustness, we apply our approach to our dataset contaminated with different levels of Gaussian noise. For each input point cloud, we add noise drawn from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, where $\mu = 0$ and $\sigma = [0.0, 1.0]$, with 1.0 being the voxel length.

Figure 7 shows the accuracy of identifying containers in the dataset. The accuracy drops from 94.44% without any noise, to 84.72% at noise level $\sigma = 0.5$, and then to 70.12% at noise level $\sigma = 1.0$. As the noise level increases, the drop in accuracy is mostly attributed to the increase in false negative. In other words, more and more containers are being misclassified as non-containers. This is due to the fact that adding noise to a container may introduce holes, which in turn result in leakage of the container and hence the wrong classification of it as a non-container.

Figure 8 shows the average angular error of the best filling directions found by our approach, at different noise levels. The error rises from 16.64° without any noise, to 38.66° at noise level $\sigma = 0.5$, and then to 40.12° at noise level $\sigma = 1.0$.

Figure 9 shows the precision and recall of the transfer direction against noise, using an angular error tolerance of 20°. Both the precision and recall drop gradually as the noise level increases. The precision drops from 0.82 without any noise, to 0.70 at noise level $\sigma = 0.5$, and then stays relatively constant.

We note that in practice, the full-view 3D data captured by a Structure Sensor (attached on an iPad) generally do not carry the high level of noise ($\sigma \geq 0.5$) like what we

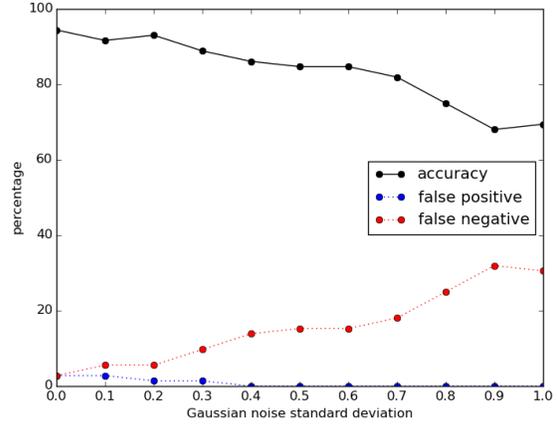


Figure 7. Accuracy of identifying containers at different noise levels.

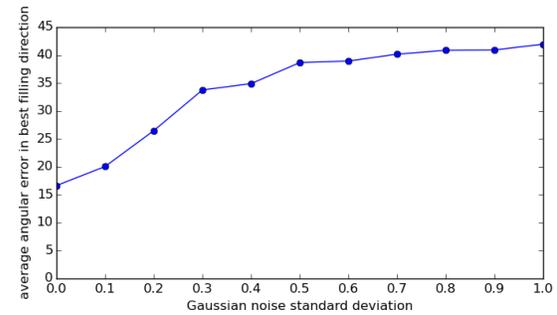


Figure 8. Average angular error in best filling directions at different noise levels.

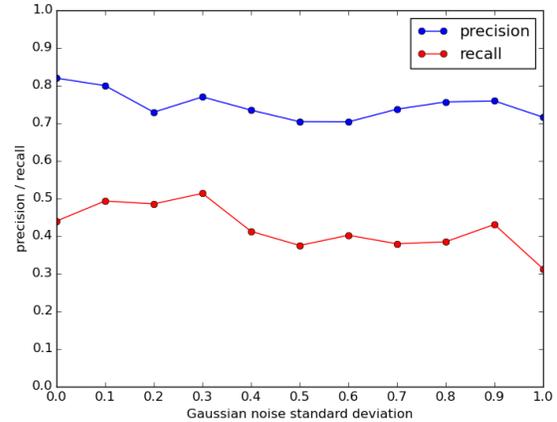


Figure 9. Precision and recall of transfer directions at different noise levels, using an angular error tolerance of 20°.

add in these synthetic experiments. The captured 3D data is generally clean and of a decent quality (see Figure 18 in the main paper for different examples), that our approach can directly apply without any denoising.

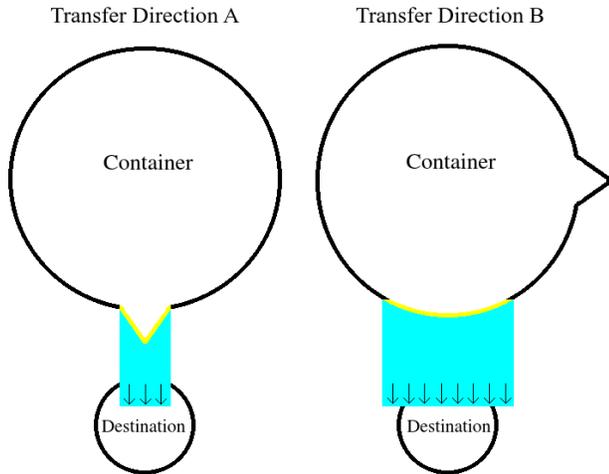


Figure 10. The flow of Transfer Direction A has a smaller cross-sectional area than that of Transfer Direction B. This enables the flow to entirely enter the destination without spillage.

4. Justification for Transfer Approach

Our criteria for choosing a desirable transfer direction rests on a few simplifying assumptions.

First we assume that minimizing spillage, *i.e.* maximizing the amount of the transferred fluid that reaches its destination, is the only objective in the transfer. For a large number of transfers (filling a wine glass, drinking a cup of water) this assumption is reasonable. For other transfers such as putting out a fire with a bucket of water, other objectives, like maximizing the flow of water transferred from the container, are more important, but our approach does not consider them.

Due to our first assumption, we assume that the destination of the transfer has a very small opening. We make this assumption because some destinations are easier to transfer to without spillage than others due to the larger size of their openings. For example, it is much easier to transfer water into a bathtub than into a test tube without any spillage. When the destination is easy, the transfer direction does not matter, so we focus on hard destinations.

We further assume that the transfer consists of a simple cylinder-shaped flow of water emerging from the container and ending at the destination. As we show in Figure 10, it is clear that the spillage is proportional to the cross-sectional area of the flow. Therefore minimizing the cross-sectional area is equivalent to minimizing the spillage. In our simplified voxel domain we can only approximately calculate the cross-sectional area.

In the real world, the shape of the flow will depend on the rate at which the container is tilted. Since we are working with a simplified physics model where we only want to consider static equilibrium situations, we simulate a con-

stant "infinitesimal" tilt rate by adding a layer of imaginary water voxels to the existing water voxels.

5. Voxelization Scheme

In this work, we use a uniform sized cubic voxel representation. Other voxel representations can be used which may result in lower quantization error at the expense of higher complexity and longer processing time [3, 4]. We decide to use the uniform cubic voxel representation for simplicity and computational efficiency.

References

- [1] G. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Springer, 2011. 9
- [2] M. do Carmo. *Riemannian Geometry*. Birkhäuser Verlag, 1992. 9
- [3] J. Ryde and M. Brünig. Non-cubic occupied voxel lists for robot maps. In *IROS*, 2009. 10
- [4] J. Ryde and M. Brünig. Lattice occupied voxel lists for representation of spatial occupancy. In *IROS*, 2010. 10