

Probabilistic Occlusion Culling using Confidence Maps for High-Quality Rendering of Large Particle Data

SUPPLEMENTARY MATERIAL

Mohamed Ibrahim, Peter Rautek, Guido Reina, Marco Agus, and Markus Hadwiger



-
- *Mohamed Ibrahim, Peter Rautek, and Markus Hadwiger are with King Abdullah University for Science and Technology (KAUST), Visual Computing Center, Thuwal, 23955-6900, Saudi Arabia. E-mail: { mohamed.ibrahim, peter.rautek, markus.hadwiger } @kaust.edu.sa.*
 - *Guido Reina is with Visualization Research Center (VISUS) at the University of Stuttgart, Germany. E-mail: guido.reina@visus.uni-stuttgart.de.*
 - *Marco Agus is with College of Science and Engineering, Hamad Bin Khalifa University, Qatar Foundation, Doha, Qatar. E-mail: magus@hbku.edu.qa.*

A SAMPLE ACCEPTANCE PROBABILITY COMPUTATION

Computing the *depth acceptance probability* as in Eq. 21 for large data sets presents a significant computational challenge. This is because the rendering of such large data sets results in many (hundreds of thousands to millions) of particles projecting to a single screen-space pixel resulting in large values of N . Therefore, the Binomial Coefficient factor of the Binomial Cumulative Distribution Function in Eq. 21 will be very costly to compute for large values of N and gets more costly as the difference between N and $k(d_s)$ increases.

There are several methods in the literature approximating the Binomial Coefficient such as Stirling's Approximation [8] resulting in a more computationally tractable computation of the Binomial Coefficient. However, even with the approximation of the Binomial Coefficient, the cumulative distribution function computation in Eq. 21 still poses a computational challenge since we are interested in the sum of the probability mass function of the Binomial distribution of a tentatively large range, namely the range from $k(d_s)$ and N .

We are therefore interested in an approximation to the Cumulative Distribution Function of the Binomial Distribution that is accurate and computationally efficient.

Normal Approximation of the Binomial Distribution Function.

For large (≥ 10) values of $N \cdot p(d_s)$ and $N \cdot p(d_s) \cdot q(d_s)$ we can approximate the Binomial Distribution Function by a Normal Distribution Function [6]. Therefore, a random variable K that follows a Binomial Distribution with parameters N and $p(d_s)$, approximately follows a Normal Distribution with a mean $\mu = N \cdot p(d_s)$ and variance $\sigma^2 = N \cdot p(d_s) \cdot q(d_s)$ as outlined in Eq. 1.

$$K \sim B(N, p(d_s)) \approx K \sim N(\mu, \sigma^2) \quad (24)$$

where $\mu = N \cdot p(d_s)$ and $\sigma^2 = N \cdot p(d_s) \cdot q(d_s)$.

We can therefore re-write the cumulative distribution function in Eq. 21 in as the cumulative distribution function of the Normal distribution as outlined in Eq. 2.

$$P(k(d_s) \leq K \leq N) = \int_{k(d_s)}^N \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{1}{2}\left(\frac{k-\mu}{\sigma}\right)^2} dk \quad (25)$$

Since the Binomial Distribution is a discrete distribution, approximating it with the continuous Normal Distribution Function requires adding a *continuity correction* term, $c = 0.5$ [6]. Adding the *continuity correction* term to Eq. 2, the Normal cumulative function would be

$$P(k(d_s) + c \leq K \leq N + c) = \int_{k(d_s)+c}^{N+c} \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{1}{2}\left(\frac{k-\mu}{\sigma}\right)^2} dk \quad (26)$$

Standard Normal Transformation We can further simplify the Approximation of the Binomial Cumulative Distribution Function in Eq. 5 if we transform the Normal Cumulative Distribution Function to a Standard Normal Cumulative Distribution Function. Note that if a random variable $Z \sim N(0, 1)$, then a random variable $X = \sigma Z + \mu \sim N(\mu, \sigma^2)$ [6]. This in turn means that $Z = (X - \mu)/\sigma \sim N(0, 1)$. Therefore, we can re-write the Cumulative Distribution Function of the general Normal Distribution in terms of the Cumulative Distribution Function of the standard Normal Distribution as:

$$P(a \leq X \leq b) = P(c \leq Z \leq d) = \int_c^d \frac{1}{\sqrt{2\pi}} \exp^{-\frac{1}{2}z^2} dz \quad (27)$$

where $c = (a - \mu)/\sigma$ and $d = (b - \mu)/\sigma$.

We can therefore re-write the Cumulative Distribution Function of the general Normal Distribution in Eq. 5 in terms of the Cumulative Distribution Function of the standard Normal Distribution for a random variable $Z \sim N(0, 1)$ as follows:

$$P(k(d_s) + c \leq K \leq N + c) = P(a \leq Z \leq b) \quad (28)$$

where $a = (k(d_s) + c - \mu)/\sigma$ and $b = (N + c - \mu)/\sigma$.

Therefore, the final formulation of the standard transformation of the Normal approximation of the Cumulative Distribution Function of the Binomial Distribution in Eq. 21 would be:

$$P(a \leq Z \leq b) = \int_a^b \frac{1}{\sqrt{2\pi}} \exp^{-\frac{1}{2}z^2} dz \quad (29)$$

where $a = (k(d_s) + c - \mu)/\sigma$ and $b = (N + c - \mu)/\sigma$.

To simplify computations further, we can rewrite Eq. 6 as

$$\begin{aligned} P(a \leq Z \leq b) &= P(Z \leq b) - P(Z \leq a) \\ &= 1 - \int_{-\infty}^a \frac{1}{\sqrt{2\pi}} \exp^{-\frac{1}{2}z^2} dz \end{aligned} \quad (30)$$

where $a = (k(d_s) + c - \mu)/\sigma$ and $b = (N + c - \mu)/\sigma$.

Computing the *depth acceptance probability* is therefore now transformed into computing a simple cumulative distribution function of the standard Normal distribution. Which can be done in a pre-processing step for a *percentile of interest* of the standard normal distribution. Denoting by z_α the *critical value* (100(1 - α)th *percentile*) of the standard normal distribution such that $P(Z \geq z_\alpha) = \alpha$, we pre-compute the range of *critical values* $[-z_\alpha, z_\alpha]$ [6]. With $\alpha = .001$, we capture 99.8% of the area under the standard normal distribution curve, and store the cumulative distribution function values over $[-z_\alpha, z_\alpha]$ on the GPU turning the *depth acceptance probability* into a simple lookup. We refer the reader to Fig. 1 for a plot of the standard normal approximation of the Binomial cumulative distribution function for different values of N and $p(d_s)$.

B EXTENDED EVALUATION

Table 1 gives a full list of data sets that we have used for the evaluation, and Table 2 and Table 3 provide the corresponding extended results for performance and for culling, respectively. Fig. 5 shows renderings of all data sets, of the three different views corresponding to the results reported in Table 2 and Table 3.

Table 5. **Data sets** used for evaluation, with particle and meshlet counts.

data set	# particles	# meshlets
copper/silver mixture	14,500,000	901,462
expanding fluid layer	30,000,000	1,880,040
covid-19	40,048,645	2,733,504
laser ablation	48,000,000	3,004,700
large laser ablation	199,940,704	12,513,900
16 \times copper/silver mixture	232,000,000	14,420,300
9 \times expanding fluid layer	270,000,000	16,916,500

C SOURCE CODE AND DATA SETS

The source code for our renderer and some data sets can be found at <http://vccvisualization.org/research/particleculling/>

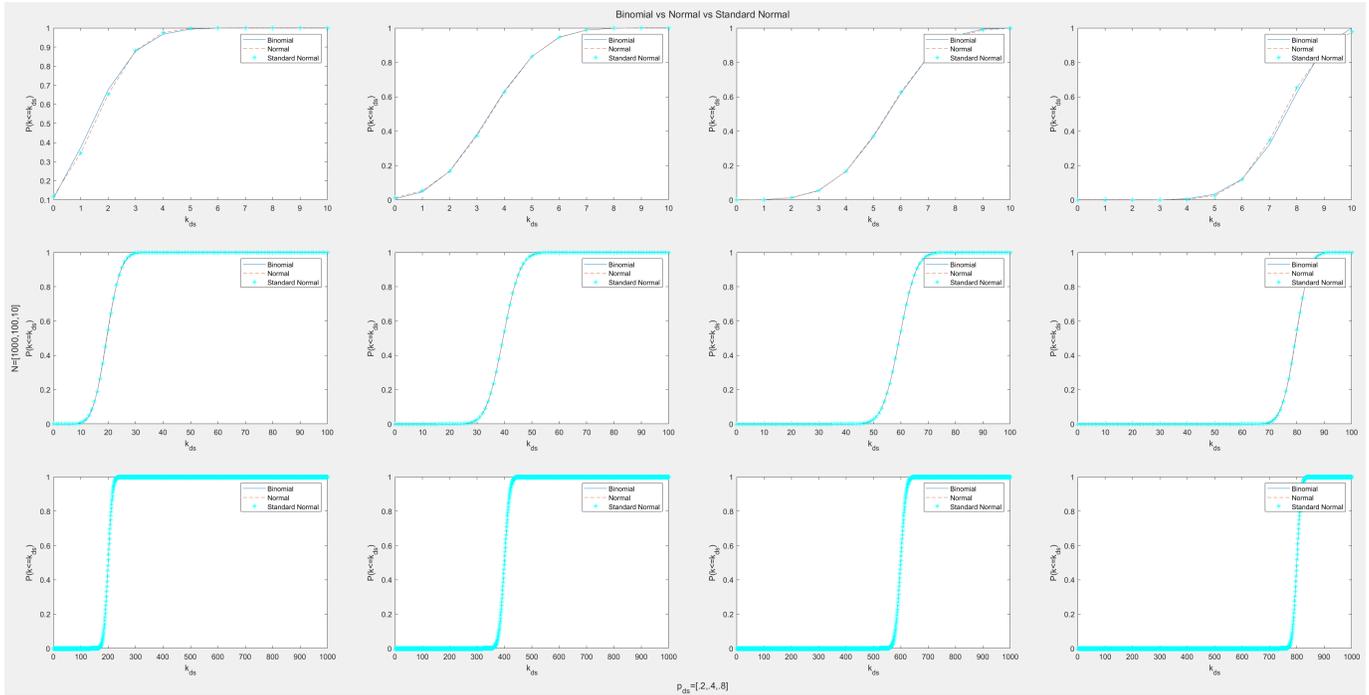


Fig. 13. **Binomial Cumulative Function Approximation** A plot of the Binomial Cumulative Function, the cumulative distribution function of the Normal approximation of the Binomial, and the cumulative distribution function of the Standard Normal equivalent of the Normal distribution. We vary N from top to bottom, and vary $p(d_s)$ from left to right. Note that the approximation is most accurate for large N .

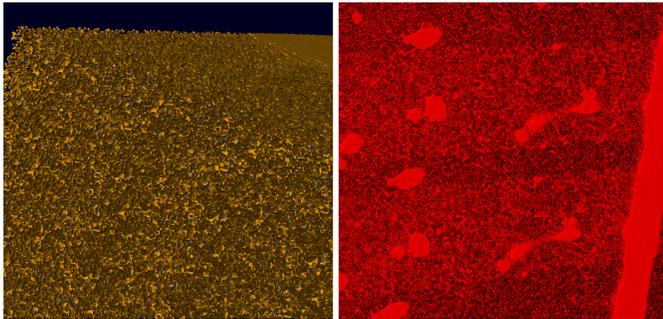


Fig. 14. Filaments and banding in the gas phase introduced by the normal smoothing from Grottel et al. [16]. Left: copper/silver mixture, right: fluid at frame 100.

D EXTENDED COMPARISON TO PREVIOUS WORK

Performance

We compared the performance of our approach against MegaMol and Grottel et al. [15] in Sect. 7 in the main text. Another standard approach for the visualization of large-scale particle data is using a particle k-D tree (P-k-d tree) [37] as a zero-overhead acceleration structure together with ray tracing (see Fig. 4). A recent improvement based on NVIDIA OptiX allows to trade-off the overhead and speed of a BVH versus the compactness and lower performance of a P-k-d tree [12]. We have tested the performance of the provided implementation with the large laser ablation using a single P-k-d tree, and it can outperform our approach without or during culling.

However, when samples per pixel are increased to allow for sub-pixel details, performance degrades linearly (as expected, see Fig. 3). Our goal of having tens to hundreds of samples can only be achieved interactively if the ray tracer uses an accumulation buffer and adds samples progressively. That way, image quality gradually increases as long as the user is not manipulating the camera. The P-k-d [12]

approach outperforms ours in the ‘dc’ stage and when we sample the data once per pixel in the ‘ac’ stage. However, after the culling computations converge, our method can generate additional samples much faster than the P-k-d [12] approach. In this example this happens after frame 22, and we report performance using hardware-supported multi-sampling for anti-aliasing (MSAA) with 2 or more samples.

By design, the overlap of sub-trees in a P-k-d tree limits its performance, as the first found hit can not early-terminate traversal. Combin-

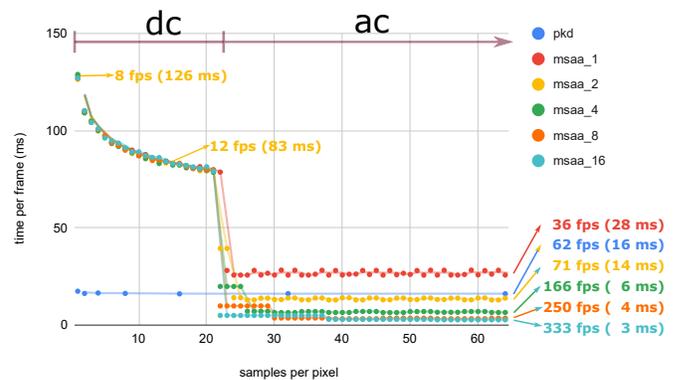


Fig. 15. **Convergence and comparison with P-k-d performance.** A comparison of the rendering performance of our approach versus the approach presented by Gralka et al. [12] using a pure P-k-d data structure and the large laser ablation data set, with increasing samples per pixel (x axis). We highlight the different stages (during culling/dc, and after culling/ac) in the plot to demonstrate the convergence behaviour of our approach. We report the rendering performance measure (fps) for the approach by Gralka et al. [12] and for our approach with MSAA enabled, with one to eight samples. In addition, we highlight the initial fps and the average fps during the dc stage for our approach with MSAA enabled with two samples.

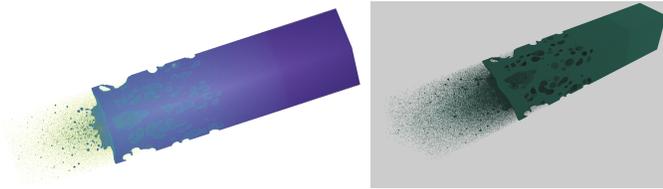


Fig. 16. **P-k-d Quality.** A rendering quality comparison of the rendering performance of our approach with MSAA with two samples versus the approach presented by Gralka et al. [12] using a pure P-k-d data structure and the large laser ablation data set, with 64 samples per pixel for a similar view of the data set (ours uses orthographic projection, while Gralka et al. [12] use perspective projection).

ing our occlusion estimate with ray traversal could provide the required information for early termination, but we leave this investigation as future work for now.

Quality

We compared the rendering quality of our approach against MegaMol and Grottel et al. [15] in Sect. 7, and in Fig. 12 in the main text. Moreover, the rendering quality is lowest with the brute-force renderer that samples each pixel once and does not apply any post processing operations to the image. It suffers from heavy aliasing, especially with very large data sets. The normal smoothing used by Grottel et al. [16] results in more pleasant images, but limits the visible features to the order of the current resolution and zoom level (normals are smoothed over a 3x3 pixel neighborhood). It is also geared towards bulk material and can thus introduce filament artifacts and banding in the gas phase, which can be mistaken for features in the data (see Fig. 2). This works contrary to our goal of preserving sub-pixel details, same as in our previous work [22], to not remove features that might indicate structures of interest only because exploration is started at an inappropriate zoom level - the same holds for level-of-detail approaches. For instance, as shown in Fig. 12, the resulting rendering of the large laser ablation data set using the super-sampling performed by our method successfully captures interesting features, marked by the dashed ellipses, that the single-sampled renderings produced by Grottel et al. [16] fail to capture.

In addition, in Fig. 4 we show the resulting rendering of the large laser ablation data set from which the performance numbers in Fig. 3 were generated. Our rendering produces a less noisy image where interesting features such as the crown are clearly visible, while the image generated by Fig. 4 captures the shadows and ambient occlusion more.

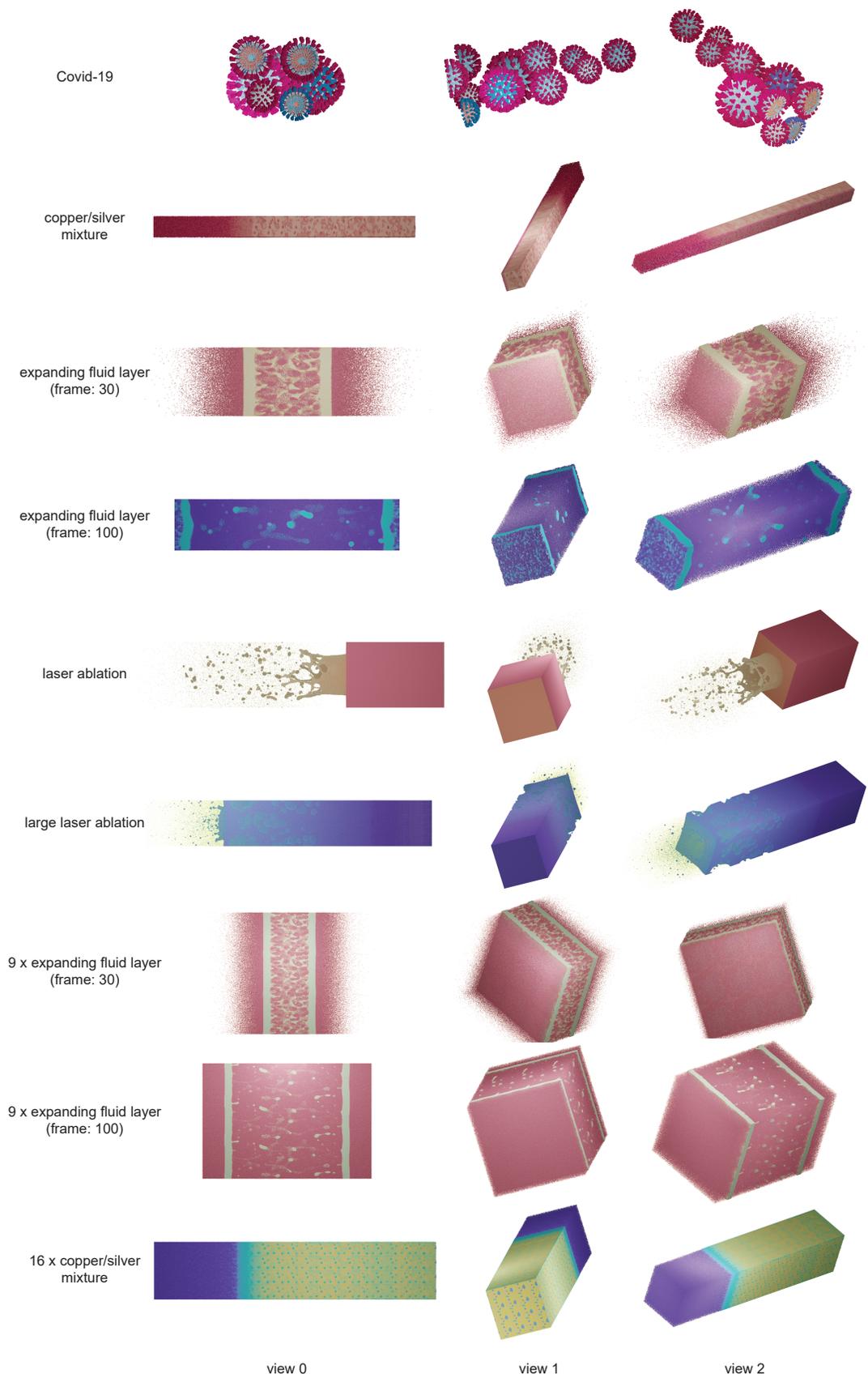


Fig. 17. **Evaluated data set views.** Visualizations of the three different views for all data sets used for measuring the numerical results given in Table 2 and Table 3.

Table 6. **Rendering performance.** For each data set, for three different views, we report average rendering speed (fps) without using occlusion culling (w/o c), during the culling process (dc), and after occlusion culling is finished (ac). For these results we used a confidence value of $C_{occ} = 0.95$, and a rendering budget $b = 2$ meshlets per node.

data set	view	fps				
		w/o c	dc	ac	MegaMol	Grottel et al.
copper/silver mixture	0	71	52	252	103	59
	1	59	53	212	185	79
	2	66	52	207	81	65
expanding fluid layer (frame: 30)	0	28	29	91	68	18
	1	33	35	118	123	14
	2	28	82	90	92	15
expanding fluid layer (frame: 100)	0	32	23	82	72	9
	1	35	30	96	108	15
	2	31	21	80	83	10
covid-19	0	19	22	56	32	18
	1	28	25	63	38	17
	2	27	24	59	50	20
laser ablation	0	50	30	115	63	200
	1	38	40	138	44	110
	2	36	40	130	59	130
large laser ablation	0	16	13	35	14	59
	1	15	18	42	9	28
	2	16	14	36	13	40
16 × copper/silver mixture	0	3	8	34	7	35
	1	6	12	41	13	12
	2	5	9	37	8	16
9 × expanding fluid layer (frame: 30)	0	4	9	27	7	5
	1	4	6	26	12	6
	2	4	9	18	7	7
9 × expanding fluid layer (frame: 100)	0	4	8	22	7	9
	1	4	8	21	12	9
	2	4	6	14	7	9

Table 7. **Culling efficiency.** We report the percentage of culled meshlets (c) and the number of samples (smp) required for culling convergence. The middle column pair is for our acceptance probability (Sect. 4.5), whereas “random” means 50:50 sample acceptance. Confidence value was $C_{occ} = 0.95$, using a rendering budget of $b = 2$ meshlets per node.

data set	view	random		probabilistic		Grottel et al. c(%)
		c (%)	smp	c (%)	smp	
copper/silver mixture	0	24	120	82	136	72
	1	20	136	75	80	87
	2	27	136	76	112	76
expanding fluid layer (frame: 30)	0	28	88	73	104	67
	1	25	88	81	96	53
	2	22	96	74	96	49
expanding fluid layer (frame: 100)	0	21	136	64	112	66
	1	13	88	68	112	70
	2	5	56	62	120	62
covid-19	0	42	200	75	112	56
	1	29	128	58	120	32
	2	25	152	60	120	24
laser ablation	0	17	88	66	56	79
	1	41	128	86	112	65
	2	42	48	83	112	68
large laser ablation	0	8	64	68	208	89
	1	36	96	79	136	79
	2	32	272	70	176	83
16 × copper/silver mixture	0	35	104	89	88	93
	1	59	104	92	72	82
	2	44	112	90	88	87
9 × expanding fluid layer (frame: 30)	0	62	72	90	72	49
	1	36	80	83	161	57
	2	54	240	88	112	44
9 × expanding fluid layer (frame: 100)	0	62	88	84	88	75
	1	51	264	83	112	73
	2	28	136	75	112	77