Science Advances
AAAS

Manuscript Template

High-speed acoustic holography with arbitrary scattering objects

Ryuji Hirayama,[1]* Giorgos Christopoulos,[1] Diego Martinez Plasencia,[1] Sriram Subramanian[1]

[1]Department of Computer Science, University College London, UK

*Corresponding author. Email: r.hirayama@ucl.ac.uk

Recent advances in high-speed acoustic holography have enabled levitation-based volumetric displays with tactile and audio sensations. However, current approaches do not compute sound scattering of objects' surfaces; thus, any physical object inside can distort the sound field. Here, we present a new technique that allows high-speed multi-point levitation even with arbitrary sound-scattering surfaces and demonstrate a volumetric display that works in the presence of any physical object. Our technique has a *two-step* scattering model and a *simplified* levitation solver, which together can achieve over 10,000 updates per second to create volumetric images above and below static sound-scattering objects. The model estimates transducer contributions in real-time by reformulating the boundary element method for acoustic holography, and the solver creates multiple levitation traps. We explain how our technique achieves its speed with minimum loss in the trap quality and illustrate how it brings digital and physical content together by demonstrating new interactive applications.

Introduction

Acoustic levitation (*1*), a technique that utilizes mechanical energy of sound to levitate and manipulate materials, has been significantly advanced over the last decade through the introduction of two fundamental techniques: phased arrays of transducers (PATs) (*2*, *3*) and acoustic holography (*4*–*6*). PATs allow dynamic control of dense arrays of sound sources (e.g., 16 × 16 ultrasound transducers) while holography, a wavefront-handling technique originally developed in optics, enabled PATs to accurately control sound fields in 3D space. Thanks to its capability of levitating almost any type of materials, acoustic holography using PATs has many potential applications in laboratory-on-chip (*7*), biology (*8*), computational fabrication (*9*), and mid-air displays (*6*, *10*–*16*). Acoustic levitation is also emerging as a strong candidate for creating new mixed-reality (MR) interfaces that can seamlessly blend the digital and physical worlds, as envisioned in the Ultimate Display of Ivan Sutherland (*17*).

In general, acoustic holography using PATs relies on a linear model (*15*, *18*, *19*), represented by using a transmission matrix $\boldsymbol{F}$. The matrix $\boldsymbol{F}$ describes how complex activations of N transducers ($\boldsymbol{\tau} \in \mathbb{C}^N$) contribute to the complex acoustic pressures at L points of interest in a sound field ($\boldsymbol{\zeta} \in \mathbb{C}^L$), using a linear system: $\boldsymbol{\zeta} = \boldsymbol{F}\boldsymbol{\tau}$, with $L \ll N$. Each element of this matrix ($F_{l,n}$) is equal to the pressure at the l-th point of interest generated by the n-th transducer when its activation is 1 (i.e., the maximum amplitude with a phase delay of 0 rad), and it can be approximated as a piston model (*20*) if we consider only direct contributions. Using this common linear model, existing approaches use different solvers to obtain the transducers' activations $\boldsymbol{\tau}$ that generate an ideal sound field $\boldsymbol{\zeta}$, which, for example, creates focal points to provide tactile sensations (*21*, *22*) or provides the maximum trapping stiffness (i.e., the Laplacian of the Gor'kov potential, commonly been used as a metric to assess how strong each acoustic trap is (*4*, *6*, *15*)) for levitating particles at desired positions (*4*). One critical milestone in this solving process for acoustic levitation was the introduction of the holographic acoustic element (HAE) framework, which simplified the computation of levitation traps by encoding them as the combination of a holographic acoustic lens creating focal points and a fixed levitation signature (*4*). This framework supports a huge range of symmetric transducer arrangements (e.g., single-sided, top-bottom, V-shape) and has been extended to multi-point levitation *(6)*. Recent algorithmic advances have further accelerated the computational speed of this framework, and consequently, the accelerated update rates (i.e., 10,000 fps) have enabled PATs to create volumetric visual content (i.e., high-speed levitation) in mid-air using the persistence of vision (POV) effect, together with tactile and audio sensations in order to provide multi-modal experiences (*13*, *15*).

However, realizing the full potential of such approaches is hindered by the model used, which operates under the assumption of empty space. That is, sound scattering of objects' surfaces is not taken into account; thus, any physical object within the working volume can distort the sound field and cause particles to fall.

Transmission matrices $\boldsymbol{F}$ only capture direct contribution from each transducer to each point, ignoring interactions with any sound-scattering objects and implicitly representing an empty working volume. The only objects permitted within the working volume are acoustically transparent materials, which are carefully chosen not to affect sound fields (*12*), along with the levitated particles, which are usually much smaller than the acoustic wavelength (e.g., $\lambda = 8.65\ mm$ in this study) and thus can be considered as acoustically transparent as well. To date, there have been limited explorations of acoustically manipulated particles in a presence of sound-scattering objects. For example, in one set of papers, the authors explored 2D plane manipulation above flat reflector (*2*, *6*, *19*, *23*), while in another approach the authors used PATs with acoustic metamaterials to demonstrate a single-particle levitation above a cloaked object (*24*).

Models such as the boundary element method (BEM (*25*, *26*)) can simulate sound-scattering fields, and BEM has been used to levitate objects several times larger than the wavelength (*27*) or to assemble nanoparticles inside arbitrary shaped closed reservoirs (*28*, *29*). However, BEM is usually considered incompatible with real-time applications, particularly for high demands of POV display applications (i.e., 10,000 fps), and no dynamic manipulation using BEM has been demonstrated in those existing works.

To make full use of acoustic holography in more flexible environments, we require a new acoustic holographic technique that does not rely on the assumption of the empty working volume and works in the presence of arbitrary sound-scattering objects. The main challenge in developing such techniques is that the entire process of both ***modeling*** the transmission matrix and ***solving*** for transducer phases must be computed in real time for practical applications of particle manipulation (e.g., 50 fps to manipulate particles at 1 cm/s with a step size of 0.2 mm). This becomes even more challenging in order to create volumetric images using the POV effect (*13*, *15*), as these require update rates above 10,000 fps. Thus, producing models as computationally efficient as transmission matrices, but with BEM's power to capture sound-scattering hence becomes the first obvious challenge.

For solvers, on the other hand, the HAE framework does not account for sound scattering and thus cannot provide the optimum solutions within the non-empty working volume (i.e., the top array with reflector, see Fig. 1). Therefore, to develop a high-performance solver without the HAE framework, we need more efficient metrics to assess trap quality, compared to the most common current metric given by trapping stiffness (i.e., Laplacian of Gor'kov potential (*4*, *6*)).

Here, we present a novel high-performance approach to modeling the extended transmission matrix and solving for transducer phases. Our technique has two novel computational components: a two-step scattering model and a simplified levitation solver. In these components, physical phenomena (i.e., sound scattering, acoustic levitation) are rebuilt or simplified as models that are suitable to be computed at high update rates. We start by reformulating BEM to pre-compute the contribution of each transducer to the mesh and then use these pre-computed values in updating the transmission matrix in real time as the trap positions move. This extended version of the transmission matrix keeps the efficiency of the empty-volume methods but provides the accuracy that is exactly equivalent to BEM. Additionally, we show that a simplified Gor'kov potential can be used as a new metric in our solver instead of stiffness, further improving the computational speed with negligible loss of accuracy. Our approach allows high-speed and accurate multi-point acoustic manipulation, even with arbitrary sound-scattering objects (see Fig. 1 and Movie S1). It allows the creation of volumetric POV images with arbitrarily shaped objects in the working volume by creating levitation traps at high computational rates. Our technique provides extra freedom in system design and allows previously impractical application scenarios, which inherently involve physical objects in their working space, such as mid-air MR displays (see Fig. 1b) and contactless manufacturing. Additionally, thanks to the high computational rates, the displayed content can be interactive to user inputs (e.g., keyboard, hand gestures) in real time. We illustrate for the first time how our acoustic holographic technique brings digital and physical content together by demonstrating several MR applications, such as a mid-air screen, a point-scanning-based volumetric display, and a surface-scanning-based volumetric display. We are the first to demonstrate a free-space surface-scanning-based volumetric display that can create full volumetric images in mid-air, within a non-empty working volume.

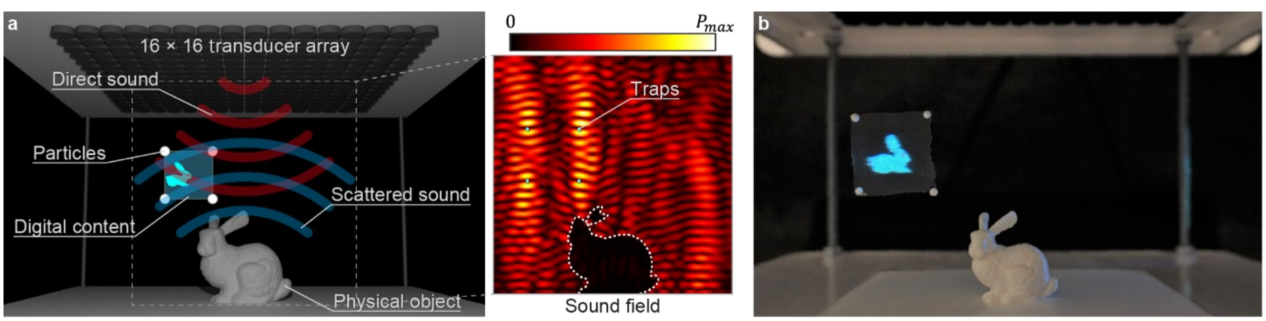

Fig. 1. Real-time acoustic holography with arbitrary scattering surfaces. (**a**) Schematical concept of our acoustic holographic technique that can create multiple levitation traps in a presence of sound-scattering physical objects. P_{max} represents the maximum amplitude of the pressure in the sound field. (**b**) Experimental example of our technique that can levitate four particles with a projection screen (i.e., a piece of light fabric), demonstrating a mixed-reality display that creates digital content in the presence of a 3D-printed physical object. The high computational rates of our approach enable the digital content to be interactive to user inputs (i.e., the levitated screen moves according to the keyboard input).

Results

Model and Solver

First, we show how our new model and solver realized high-speed multi-point levitation with minimum loss of accuracy, even within a non-empty working volume.

Two-step scattering model: BEM can model sound scattering objects by modeling them as a mesh of M boundary elements (i.e., we use meshes with 3,000–6,000 elements in our examples). A transmission matrix $\boldsymbol{E}$ that captures both the direct and scattering contributions of the transducers to target points could be computed by repeating the BEM computation for each of the N single transducers (i.e., $N = 256$). However, each BEM computation involves solving a large dense linear equation system, and therefore, repeating this process for every transducer in real time is not practical. Thus, we propose a new technique to reformulate BEM for acoustic holography to define the matrix $\boldsymbol{E}$ using three matrices as: $\boldsymbol{E} = \boldsymbol{F} + \boldsymbol{GH}$ (see Fig. 2a). Here, the matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ represent the respective contributions from the transducers and mesh elements to the points of interest (i.e., thus, $\boldsymbol{F}$ is the conventional transmission matrix capturing only direct contributions), and the matrix $\boldsymbol{H}$ represents the contribution from the transducers to the mesh elements. The sizes of these matrices are $L \times N$ for $\boldsymbol{E}$ and $\boldsymbol{F}$, $L \times M$ for $\boldsymbol{G}$, and $M \times N$ for $\boldsymbol{H}$. Given the fact that the inequality $L \ll N \ll M$ is usually satisfied in acoustic levitation, the determination of $\boldsymbol{H}$ is more time-consuming than the other matrices.

For static set-ups, $\boldsymbol{H}$ is constant and we can thus pre-compute it once the set-up is defined (i.e., position and normal of each transducer and position, area and normal of each mesh element in the reflector). In contrast, computing $\boldsymbol{F}$ and $\boldsymbol{G}$ requires the positions of points of interest in addition to the set-up information. For interactive applications, these points of interest are usually unknown beforehand, and thus $\boldsymbol{F}$ and $\boldsymbol{G}$ need to be created in real time depending on the application logic and/or user input. While $\boldsymbol{H}$ must be pre-computed, computation of $\boldsymbol{F}$ and $\boldsymbol{G}$ is highly parallelizable, and our model can achieve high computational rates for this modeling process by using a graphics processing unit (GPU) even in the presence of static sound-scattering objects. Fig. S6a shows the computational speed of only this modeling process after the pre-computation part. Note here that our model is exactly equivalent to BEM, not relying on any approximation to compute the acoustic pressures on the meshes, and thus can be used to model any geometry of scattering objects without sacrifice in accuracy, unlike the methods based on the Rayleigh integral (*18*, *19*, *30*), which are limited to flat or slightly curved reflectors. We also note that our model does not require high sampling resolution for 3D models' mesh (i.e., the best-balanced mesh size is $\lambda/2$, see *Mesh-size Dependency of the Trap Quality* and Fig. S7). We also discuss how to adapt our approach to dynamically changing meshes later in *Discussion*.

Simplified levitation solver: We propose a simplified solver using the model above. Our *SIMPLIFIED* solver uses a gradient descent minimizing a simplified metric U' at every trap position $\boldsymbol{r_j} = (x_j, y_j, z_j)$, allowing us to create multiple stable traps at high computational speed. Our metric U' is based on the Gor'kov potential U, which can be used to compute the acoustic radiation force $\boldsymbol{F^{rad}}$ applied on a small particle (i.e., much smaller than the acoustic wavelength) at the point j as: $\boldsymbol{F^{rad}} = -\boldsymbol{\nabla} U(\boldsymbol{r_j})$. Here, $U(\boldsymbol{r_j})$ can be determined by the complex acoustic pressure p and its spatial derivatives at the trap position $\boldsymbol{r_j}$ and constant values (K_1 and K_2) as (*31*):

$$U(\boldsymbol{r_j}) = K_1|p|^2 - K_2\left(\left|\frac{\partial p}{\partial x}\right|^2 + \left|\frac{\partial p}{\partial y}\right|^2 + \left|\frac{\partial p}{\partial z}\right|^2\right). \tag{1}$$

Trapping stiffness (*4*, *6*) is a common metric to evaluate (and optimize) the quality of acoustic traps and is computed as the Laplacian of the Gor'kov potential ($\nabla^2 U$) at the point j. A traditional method is to create levitation traps by maximizing such trapping stiffness at the desired locations, with an optimization algorithm such as gradient descent (*4*). However, computing stiffness $\nabla^2 U(\boldsymbol{r_j})$ requires sampling pressure values at many points of interest around each trap and thus is computationally heavy for use in real-time applications.

In this study, we accelerate our solving process of creating J traps by using a simplified Gor'kov potential $U'(\boldsymbol{r_j})$ as a new metric (i.e., our cost function in gradient descent):

$$U'(\boldsymbol{r_j}) = K_1|p|^2 - K_2\left|\frac{\partial p}{\partial z}\right|^2. \tag{2}$$

The advantage of this new metric is that it can be computed by sampling pressure values at only two points per trap (i.e., the number of total points of interest is $L = 2J$). This simplified metric is suitable for our experimental set-ups, in which the transducers face downward (i.e., $-z$ direction) and sound-scattering objects are placed underneath (see Fig. 1a), allowing them to create standing-wave-like acoustic traps along the z-axis, similar to the commonly-used top-bottom set-ups (*6*).

Our simplification in Eq. 2 approximates sufficiently the potential $U(\boldsymbol{r_j})$ because the derivative of the pressure along the z-axis is more dominant than the derivatives along the other axes. Also, the Gor'kov potential along the z-axis behaves locally as a sinusoidal pattern (*32*). Thus, the second derivative of such sinusoidal pattern (i.e., trapping stiffness) should also be sinusoidal of opposite sign, supporting our assumption that a negative relationship between $U(\boldsymbol{r_j})$ and its Laplacian ($\nabla^2 U(\boldsymbol{r_j})$) still holds. Figure 2b validates this, showing the relationship between our new metric and trapping stiffness in our set-ups with a very good correlation (i.e., $R^2 = 0.940$) and experimentally evaluating our assumption. Note here that our simplified metric (Eq. 2) could not be directly used in set-ups, where this assumption is not valid, but this can be easily adjusted to other set-ups such as single-sided, top-bottom, and V-shape, as shown in the section *Metric validity* and Figs. S4.

Validation and Performance Evaluation

To evaluate our *SIMPLIFIED* solver in terms of trapping stiffness, we compared it with the other two solvers, which we refer to as *BASELINE* and *HEURISTIC*. The *BASELINE* solver is a traditional method that uses a physically accurate and broadly accepted metric (i.e., trapping stiffness $\nabla^2 U(\boldsymbol{r_j})$) to optimize trapping quality (*4*) but is slow. The *HEURISTIC* solver is an extension of the HAE framework, enabling us to create traps by creating two focal points around each trap with a π-radian offset in the target phases (*6*). Although this approach is fast and would work well in a single-point manipulation, destructive interference between traps is likely to occur in multi-point manipulation (*15*). Note here that all the three solvers use our two-step scattering model.

As shown in Fig. S9, our *SIMPLIFIED* solver avoids destructive interference between multiple traps when compared to the *HEURISTIC* solver, while achieving similar quality (i.e., trapping stiffness) than the *BASELINE* solver that directly maximized $\nabla^2 U(\boldsymbol{r_j})$ (see *Comparison between the Solvers* for more detailed evaluations). Additionally, with an appropriate initialization, our *SIMPLIFIED* solver can converge within 100 iterations (see *Convergence and Initialization* and Fig. S8). Therefore, our solver represents solutions being the most balanced, realizing accurate and fast acoustic manipulation.

Figure 2c summarizes the computational performance of our acoustic holographic technique. We evaluated how the numbers of traps (J) and mesh elements (M) influence the computational speed. Here, the number of transducers (N) and the number of iterations (K) in the solver were fixed (i.e., $N = 256$, $K = 100$). The results show the linear relationship between them as expected, and high update rates over 10,000 fps (i.e., less than 0.1 ms computational time) can be achieved in several scenarios (e.g., $J = 4$ with $M = \sim 8{,}000$). For example, the 3D model of the bunny and the flat reflector (i.e., 12 × 12 cm^2), which was used in the four-trap application in Fig. 1b, is composed of 4,134 elements in total, achieving over 15,000 fps. The plots also show that even with the slowest scenario in the plots (i.e., $J = 16$ and $M = 32{,}000$), we can still get over 700 fps, which is enough to manipulate particles in real time. Although the set-up-related part cannot be computed in real-time (see *Computational Performance*), this part can be pre-computed once the set-up is defined.

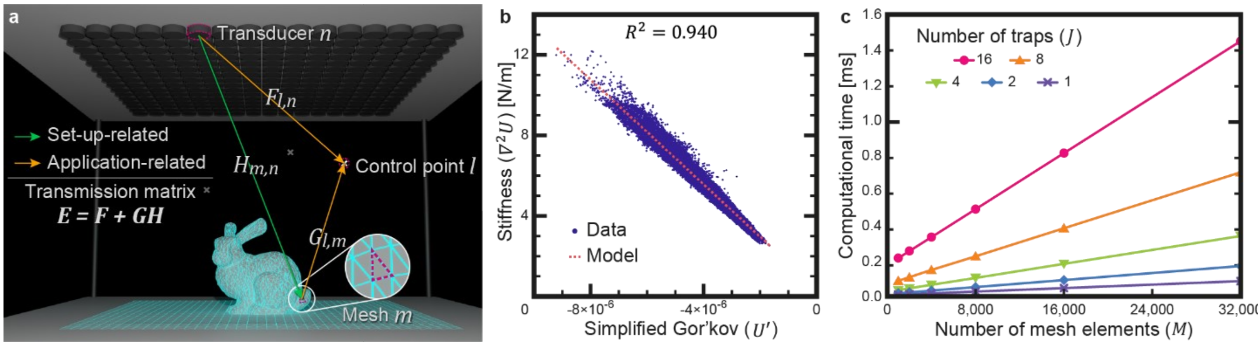

Fig. 2. Performance of the proposed technique. (a) Schematical explanation of our *two-step* scattering model, adapting the boundary element method (BEM). **(b)** Correlation between trapping stiffness $\nabla^2 U$ and the simplified form of the Gor'kov potential U', justifying the use of U' as our metric. **(c)** Computational performance of our acoustic holographic technique after pre-computation, depending on the numbers of mesh elements (M) and traps (J).

Versatile Manipulation Capabilities

The combination of our two-step scattering model and the simplified levitation solver allows real-time manipulation of materials in 3D space, in the presence of sound-scattering objects. Figure 3a shows an experimental example of levitating 10 expanded polystyrene (EPS) particles above a 3D-printed smooth surface. The simulated sound field in the xy-plane $\lambda/4$ above the trap positions (i.e., the inserted image in Fig. 3a) shows 10 high-pressure points. The closest previous demonstrations (*2*, *6*, *19*, *23*) of this example were limited to 2D plane manipulation of EPS particles or liquid droplets just above flat reflector surfaces without any scattering object. In our case, we have enabled acoustic 3D manipulation even with a non-flat reflector. In addition, particles can be levitated under sound scattering obstacles, which occlude most direct sound contributions from the transducers (see Fig. 3b), showing manipulation capabilities in scenarios that were not previously possible.

Unlike other levitation techniques such as electromagnetics, the acoustic approach can levitate almost any type of material, including solids and liquids (*1*). Figure 3c shows the manipulation of a water droplet in the presence of 3D-printed cacti. Acoustic manipulation of liquid droplets is particularly challenging, as the acoustic velocity of air particles at the trap position needs to be carefully adjusted, keeping it within the range determined by the droplet's radius and surface tension to avoid droplet atomization (*2*, *33*). The fast computational rates of our technique enable us to estimate the acoustic velocity in real time, dynamically adjusting the transducers' amplitudes to make the acoustic velocity constant along the manipulation path (see Fig. S12). Additionally, by modulating the amplitudes of all the transducers at certain frequencies, we can induce oscillatory vibrations to levitated droplets, which is useful for mixing multiple materials in a contactless manner without any cross-contamination (*34*). Furthermore, our scattering model works even if the scattering surfaces are liquids. Figure 3d shows the manipulation of a mixture of water droplets, taking into account the liquid surface of a container filled with water (see also Movie S2). We approximated the liquid surface is acoustically rigid (i.e., $\beta_m = 0$), still showing correct droplet manipulation. Such material independence lends versatility to our technique, which can be applied in fields such as computational fabrication, laboratory-on-a-chip, and biomedical imaging. The use of other β_m values is also possible, as detailed in *Two-step Scattering Model*.

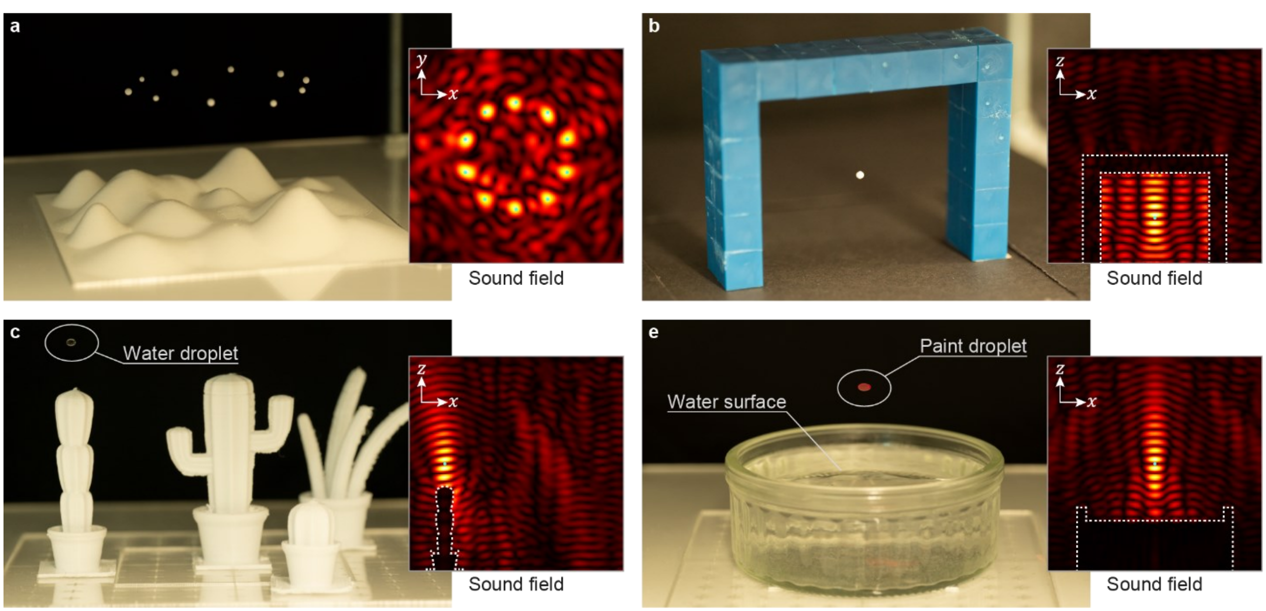

Fig. 3. Levitation capabilities of the proposed technique. (a) Our technique can create and manipulate multiple traps individually (i.e., there are 10 traps in the photograph). **(b)** Traps can be created even under sound-scattering obstacles by

utilizing scattered waves. **(c)** Materials that can be manipulated in mid-air include both solids and liquids (i.e., a water droplet is levitated). **(d)** Our scattering model works on scattering surfaces of liquids as well. The inserted boxes show simulated sound fields in the xy-plane $\lambda/4$ above the trap positions for (a) and the xz-plane on the trap positions for the others (b-d), normalized using the maximum amplitude. The white dashed lines in these figures represent the positions of the scattering objects in the planes.

Creation of POV Images Using High Update Rates

An important aspect of our technique is its computational speed. As discussed in the literature (*13*, *15*), high update rates for PATs, of ideally more than 10,000 fps, enable us to manipulate EPS particles at fast velocities (i.e., maximum velocity of 8.75 m/s with the top-bottom setup was reported in (*13*)), allowing the creation of mid-air volumetric images using the POV effect by scanning particles in 0.1 s (*35*). The fast computational speeds of our technique (see Fig. 2c) allow such a point-scanning-based method to create volumetric POV images even in the presence of sound-scattering objects. Additionally, thanks to the high update rates of our approach, created POV images can be interactive to user inputs (e.g., keyboard, hand gestures) in real time. Figure 4a shows the creation of a butterfly flapping around a 3D-printed bunny ($M = 4{,}134$), which can be controlled by hand gestures (see Movie S4), by using a single particle colored by full-color LEDs. Other examples of volumetric shapes are shown in Fig. 4b (see also Movie S3), showing two particles on top of plastic bricks ($M = 5{,}010$), while Fig. 4c shows a single particle under sound-scattering obstacles ($M = 3{,}792$). These are the first demonstration of the creation of digital volumetric images with physical objects as a new MR human–computer interface, blurring the boundary between the digital and physical worlds.

However, the volumetric geometries that the point-scanning-based approach can create are limited to simple shapes, as demonstrated in Figs 4a, 4b, and 4c, because particles must scan all the geometries in the POV time (i.e., 0.1 s). Therefore, here we additionally demonstrate for the first time a free-space surface-scanning-based display within a non-empty working volume, to create more complex volumetric shapes with many voxels (volume elements). In this approach, we levitated a piece of light fabric with the same levitation set-up used for the point-scanning approach and used a high-speed projector (i.e., 1,440 fps) and a mirror, as shown in Fig. 4d. Our technique can rotate the fabric in the presence of sound-scattering objects at five rotations per second while synchronously projecting cross-sectional images of a 3D model on the rotating fabric, revealing the full volumetric image in mid-air due to the POV effect. The reason we used the mirror is to project images even when the projection direction and the fabric are in parallel. The two photographs taken from the different perspectives (see Figs. 4e and 4f) show the digital 3D image of a bunny projected onto the rotational fabric. The digital 3D image was created on top of a physical bunny ($M = 4{,}134$), which was 3D-printed using the same 3D model for the digital bunny. We can confirm that our system can project complex volumetric shapes in mid-air, which can be viewed from any direction.

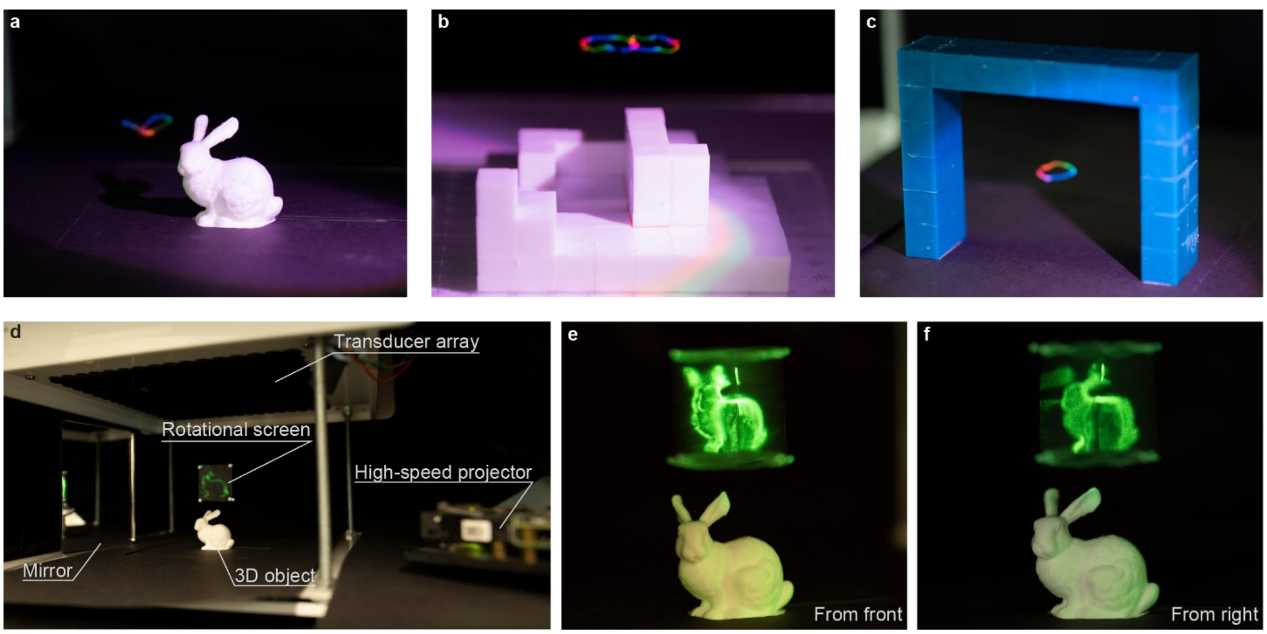

Fig. 4. Mixed-reality applications using high-speed acoustic holography. (a, b) Examples of the creation of volumetric POV images using single and multiple particles with sound-scattering objects. **(c)** POV images can be created even under sound-scattering obstacles. **(d)** Full-volumetric projection of 3D digital content together with a 3D printed object using a quickly rotating screen (i.e., five rotations per second) and a high-speed projector. **(e, f)** Two photographs taken from two different perspectives (i.e., from front and right) to demonstrate full volumetric projection. Note here that the digital and physical objects both used the same 3D model (i.e., bunny) with the same orientation.

Discussion

Prior to this work, 3D manipulations of materials using acoustic holography have been accomplished only in an empty volume. This limitation has so far forced the technology to be used in limited scenarios (i.e., no scattering objects around). Here, we overcome this limitation by reformulating and simplifying the model and solver for acoustic holography. Our approach extends the possibilities of acoustic levitation, enabling 3D printing for contactless manufacturing and mixing of physical and digital artifacts for novel MR applications. In this study, we assumed only sound-scattering objects with high acoustic impedance compared to air (e.g., plastic, water), within a single propagation medium (i.e., air). However, BEM can also be used to compute sound scattering from sound soft boundaries, even through multiple media. The same two-step approach could be applied to such more complex scenarios, accelerating computational speed and paving the way for real-time exploitation beyond the environments demonstrated.

This range of potential scenarios will also increase as we relax our current limitation of using only static scattering objects (i.e., a single pre-computed matrix $\boldsymbol{H}$), but so do the challenges that need to be considered. That is, by removing the need for an empty volume, our current method already enables ultrasound-based solutions to be applied to many more real-world settings, such as inside appliances or in the dashboard of a car.

An obvious step to support dynamic (i.e., moving/changing) objects would be to pre-compute different $\boldsymbol{H}$ matrices, one per state of the object. This would require us to know in advance the nature of the dynamic evolution of the object, but even this simple step would be enough to enable many novel applications such as 3D printing and contactless assembly, as in all these cases the evolution of the geometry is known ahead of time.

Moving towards fully interactive scenarios opens new challenges and possibilities. For objects interactively changing position and orientation but with fixed shape, the LU decomposition technique discussed in *Moving Sound-scattering Objects* and Fig. S10 could allow matrix $\boldsymbol{H}$ to be computed in real-time. The most challenging scenario is when the objects change their shapes, positions, and orientations in an unpredictable manner (e.g., an MR application where users' hands interact inside the working volume). New approaches to compute $\boldsymbol{H}$ in real time would be required here, but one potential solution is to exploit the local nature of changes. That is, if the positions and/or geometries of objects do not change drastically between updates, the solution for the previous geometry can be used as good initial estimations for the next geometry, reducing the computational cost. It is worth noting that the computational rates for this set-up-related part do not need to achieve 10,000 fps, and more conventional rates could suffice (e.g., >30 fps).

Also, our two-step scattering model can be adapted to various PAT arrangements (e.g., top-bottom, V-shape, single-sided; see Figs. S4d, S4e, S4f) with no modification. This offers great flexibility in designing new applications using our acoustic holographic technique. However, we need to note that the simplified metric should not be used as in Eq. 2 by default, but rather be adjusted to the geometric relationship between the involved PATs and trap positions (see Figs. S4a, S4b, S4c). This suggests that dynamically tuning the most suitable metric simplification for the set-up and content used would enable us to always bring the best accuracy and speed out of the device.

The point-scanning-based approach has been adopted and explored to realize free-space volumetric displays by using several levitation techniques such as acoustic (*13–15*), photophoretic (*36*), and electromagnetic traps (*37*). In this study, we for the first time introduced the surface-scanning-based approach into these levitation techniques and achieved the free-space volumetric display that can represent more voxels with minimum constraint in voxel arrangement, compared to the point-based ones (detailed in *Surface-scanning-based Volumetric Display*). In comparison with the volumetric displays using mechanically-rotated screens or emitters (*38*, *39*), the advantage of our approach is that we can manipulate the rotational screen itself within the space that the user can directly access, highlighting the MR aspect of the acoustic holographic technique proposed in this study.

Materials and Methods

<u>Modeling sound-scattering for acoustic holography</u>

Our scattering model is based on BEM (*25*, *26*). Therefore, we first describe how the conventional BEM works for general scattering problems and then how we reformulated BEM for acoustic holography in two steps, to achieve the high update rates.

Conventional BEM for Scattering Problems: In BEM, acoustic pressure at some point $\boldsymbol{x}$ can be represented as a boundary integral equation (i.e., Helmholtz-Kirchhoff integral equation) obtained via Green's theorem. In scattering problems, BEM can be computed by discretizing the surface of the scattering objects into M mesh elements. The size of the elements is small enough so that the pressure across each mesh p_m can be considered as constant across the element. Then, under certain impedance boundary conditions parametrized by β_m, the complex pressure $p(\boldsymbol{x})$ in the domain of propagation (i.e., the region in which the wave propagates) is given by the direct incident contributions $p^{inc}(\boldsymbol{x})$ and scattered contributions from every mesh element as:

$$p(\boldsymbol{x}) = p^{inc}(\boldsymbol{x}) + \sum_{m=1}^{M} p_m s_m \left[ik\beta_m G(\boldsymbol{x}_m, \boldsymbol{x}) + \frac{\partial G(\boldsymbol{x}_m, \boldsymbol{x})}{\partial n(\boldsymbol{x}_m)} \right]. \tag{3}$$

Here, s_m represents the surface area; k is the wavenumber; and β_m denotes the relative surface admittance at the boundary, computed as the ratio of acoustic impedances of the propagation medium Z_0 and the scattering object Z_s (i.e., $\beta_m = Z_0/Z_s$; $\beta_m = 0$ when the surface is acoustically rigid). $G(\boldsymbol{y}, \boldsymbol{x})$ is the so-called free-field Green's function, defined in the 3D case by:

$$G(\boldsymbol{y}, \boldsymbol{x}) = -\frac{e^{ikd(x,y)}}{4\pi d(\boldsymbol{x}, \boldsymbol{y})}. \tag{4}$$

Here, $d(\boldsymbol{x}, \boldsymbol{y})$ is the Euclidean distance between two points $\boldsymbol{x}$ and $\boldsymbol{y}$. In Eq. 3, $\partial/\partial n$ denotes the normal derivative on the boundary (i.e., the rate of increase in the direction of the mesh's normal $\boldsymbol{n}_m$). Let $\psi(\boldsymbol{x}, \boldsymbol{y})$ denote the angle between the mesh's normal at $\boldsymbol{y}$ and the vector $\boldsymbol{x} - \boldsymbol{y}$ and $\boldsymbol{\nabla}_y$ denote the gradient for the components of $\boldsymbol{y}$. The normal derivative of the Green's function at $\boldsymbol{y}$ can be represented as:

$$\frac{\partial G(\boldsymbol{y}, \boldsymbol{x})}{\partial n(\boldsymbol{y})} = \boldsymbol{n}(\boldsymbol{y}) \cdot \boldsymbol{\nabla}_y G(\boldsymbol{y}, \boldsymbol{x}) = \frac{e^{ikd(x,y)}}{4\pi d(\boldsymbol{x}, \boldsymbol{y})} \left(ik - \frac{1}{d(\boldsymbol{x}, \boldsymbol{y})} \right) \cos\psi(\boldsymbol{x}, \boldsymbol{y}). \tag{5}$$

On the other hand, when the surface is smooth around $\boldsymbol{x}_m$, the acoustic pressure on each mesh p_m can be derived from the Helmholtz-Kirchhoff integral equation under the same impedance boundary condition (*25*) as:

$$\frac{1}{2} p_m = p_m^{inc} + \sum_{\substack{m'=1 \\ m' \neq m}}^{M} p_{m'} s_{m'} \left[ik\beta_{m'} G(\boldsymbol{x}_{m'}, \boldsymbol{x}_m) + \frac{\partial G(\boldsymbol{x}_{m'}, \boldsymbol{x}_m)}{\partial n(\boldsymbol{x}_{m'})} \right];\ m = 1, \dots, M. \tag{6}$$

Eq. 6 leads a set of M linear equations to determine the M unknown pressure values at the mesh elements p_m. The equation can be represented as a simple equation system $\boldsymbol{Ap}=\boldsymbol{b}$, where each element of the matrix $\boldsymbol{A}$ and the vector $\boldsymbol{b}$ are given by:

$$b_m = p_m^{inc}. \tag{7}$$

$$A_{m,m'} = \begin{cases} 0.5, & m=m' \\ -s_{m'}\left[ik\beta_{m'}G(\boldsymbol{x}_{m'},\boldsymbol{x}_m)+\dfrac{\partial G(\boldsymbol{x}_{m'},\boldsymbol{x}_m)}{\partial n(\boldsymbol{x}_{m'})}\right], & m\neq m' \end{cases}. \tag{8}$$

Once the set of pressure values at the mesh elements ($\boldsymbol{p}=[p_1 \quad \cdots \quad p_M]^T$) is obtained by solving this equation system, we can compute sound pressure $p(\boldsymbol{x})$ at any position in the propagation field using Eq. 3. The matrix $\boldsymbol{A}$ depends only on the geometry of the boundary, while the vector $\boldsymbol{b}$ depends on the incident wave (i.e., direct sound contributions from the transducers). It must be noted that solving this equation takes a huge amount of time and memory for a large M.

Two-step Scattering Model: To compute the transmission matrix $\boldsymbol{E}$ at high update rates, our model reformulates BEM into two parts: the set-up-related and the application-related parts. Each element of the matrix $E_{l,n}$ equals the pressure $p_{l,n}$ that the n-th transducer generates at the l-th point with a transducer's complex activation $\tau_n=1$. In this study, we assumed $\beta_m=0$ in Eqs. 3 and 8 for all the sound-scattering surfaces used (i.e., plastic, water) because their acoustic impedances are very high when compared to air. Then, $p_{l,n}$ can be represented by using BEM as:

$$p_{l,n} = p_{l,n}^{inc} + \sum_{m=1}^{M} p_{m,n}s_m \frac{\partial G(\boldsymbol{x}_m,\boldsymbol{x}_l)}{\partial n(\boldsymbol{x}_m)}. \tag{9}$$

Here, $p_{l,n}^{inc}$ denotes the direct contribution from the n-th transducer to the l-th point, and $p_{m,n}$ denotes the pressure at the m-th mesh generated by the n-th transducer. Then, as shown in Fig. 2a, the transmission matrix $\boldsymbol{E}$ can be represented as:

$$\boldsymbol{E} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,N} \\ \vdots & \ddots & \vdots \\ p_{L,1} & \cdots & p_{L,N} \end{bmatrix} = \boldsymbol{F}+\boldsymbol{GH}. \tag{10}$$

$$F_{l,n} = p_{l,n}^{inc};\, G_{l,m} = s_m\frac{\partial G(\boldsymbol{x}_m,\boldsymbol{x}_l)}{\partial n(\boldsymbol{x}_m)};\, H_{m,n}=p_{m,n}. \tag{11}$$

The direct incident contribution $p_{l,n}^{inc}$ can be represented as: $p_{l,n}^{inc}=P_{l,n}\Phi_{l,n}$, where $P_{l,n}$ denotes the scalar directivity of our sound sources approximated as a piston model, and $\Phi_{l,n}$ denotes the complex phase propagation approximated as a spherical sound source:

$$P_{l,n} = \frac{2J_1(kr\sin\theta(\boldsymbol{x}_l,\boldsymbol{x}_n))}{kr\sin\theta(\boldsymbol{x}_l,\boldsymbol{x}_n)}\frac{P_{ref}}{d(\boldsymbol{x}_l,\boldsymbol{x}_n)};\; \Phi_{l,n}=e^{ikd(\boldsymbol{x}_l,\boldsymbol{x}_n)}. \tag{12}$$

Here, P_{ref} represents the transducer's reference pressure at 1 m distance; r represents the transducer's radius; $\theta(\boldsymbol{x}_l,\boldsymbol{x}_n)$ is the angle between the transducer's normal and point l; and J_1 represents a Bessel function of the first kind.

As we already mentioned, we assumed $\beta_m=0$ for all the sound-scattering surfaces in this study. The extension to other values of β_m is also possible by keeping the term of $ik\beta_{m'}G(\boldsymbol{x}_{m'},\boldsymbol{x}_m)$ in Eq. 8 when solving the matrix $\boldsymbol{H}$ and adjusting Eq. 11 to have the term when computing the matrix $\boldsymbol{G}$. We can adopt this extension, without much increasing the computational complexity.

The important point is that the matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ depend on point positions while the matrix $\boldsymbol{H}$, the largest and most computationally expensive element in our model, does not. Therefore, once the geometry of the set-up (i.e., transducers and scattering objects) is determined, $\boldsymbol{H}$ remains constant and does not have to be computed every time we update the trapping positions (i.e., the set-up-related part). On the other hand, we must compute $\boldsymbol{F}$ and $\boldsymbol{G}$ every time for interactive applications (i.e., the application-related part), but the computations of these have direct expressions given in Eq. 11 and thus are highly suitable for computing in parallel using a GPU. Therefore, once we pre-compute the matrix $\boldsymbol{H}$, the whole matrix can be computed at very high rates (see Fig. S6a). The pre-computation process for the set-up-related part to calculate the matrix $\boldsymbol{H}$ is as follows:

1. Given the geometry of the sound-scattering objects, build an $M\times M$ matrix $\boldsymbol{A}$ using Eq. 8.
2. Build an M vector $\boldsymbol{b}^{(n)}$ for the n-th transducer: $b_m^{(n)}=P_{m,n}\Phi_{m,n}$.
3. Solve $\boldsymbol{Ap}^{(n)}=\boldsymbol{b}^{(n)}$ to obtain $\boldsymbol{p}^{(n)}$ and store the results: $H_{m,n}=p_m^{(n)}$.
4. Repeat the steps 2 and 3 for all the N transducers.

In this study, we used a MATLAB function *gmres*, which uses the generalized minimum residual (GMRES) algorithm (*40*), to solve the linear systems in the step 3. An alternative way to represent the steps 2–4 is as $\boldsymbol{AH}=\boldsymbol{B}$, where $\boldsymbol{B}=[\boldsymbol{b}^{(1)} \quad \cdots \quad \boldsymbol{b}^{(N)}]$. We could also decompose the matrix $\boldsymbol{A}$ (e.g., LU decomposition) to compute $\boldsymbol{H}$ at higher speeds, instead of using GMRES.

Sound-field Simulation Using Our Model: As the conventional BEM, our model can be used for the general purpose of simulating sound fields, even though the main purpose of developing it in this study was to solve for the transducers' activation $\boldsymbol{\tau}$ to create multiple traps at high speeds. Figs. S1a and S1b show the sound fields simulated by the conventional BEM (see *Conventional BEM for Scattering Problems*) when we created single traps at different positions, while Figs. S1c and S1d were simulated by our model (see *Two-step Scattering Model*) when we used the same transducers' activations $\boldsymbol{\tau}$ as Figs. S1a and S1b, respectively. In these simulations, we used the bricks object shown in Fig. S3. We can confirm that the sound fields generated by our model are equivalent to the ones simulated by the conventional BEM.

The conventional BEM requires solving the linear equation $\boldsymbol{Ap}=\boldsymbol{b}$ every time to simulate sound fields with different $\boldsymbol{\tau}$, even with the same set-up (e.g., as in the case of Figs. S1a and S1b). In contrast in our model, once we compute the transmission matrix $\boldsymbol{E}$, it can be used for simulating sound fields with different $\boldsymbol{\tau}$ unless the same set-up is used. Note here that $\boldsymbol{E}$ can be computed at very high speeds (see Fig. S6a), once we obtain the data from the pre-computation (i.e., the matrix $\boldsymbol{H}$). Our model is especially useful for simulating and evaluating sound fields many times with different $\boldsymbol{\tau}$ but with the same set-up. Therefore, in this paper, we used our model for every evaluation and visualization of the sound fields.

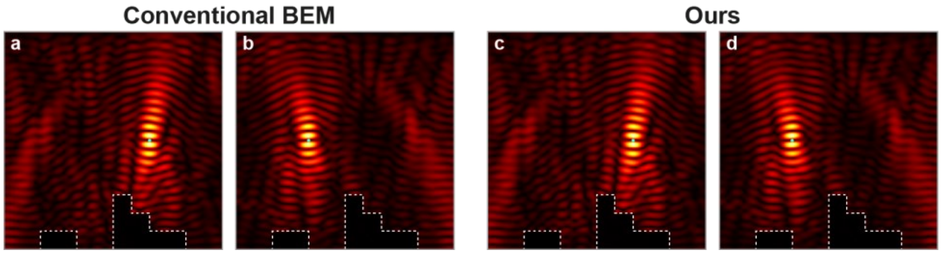

Fig. S1. Comparison between the conventional BEM and our two-step scattering model. Sound fields simulated by using **(a, b)** the conventional BEM and **(c, d)** our two-step scattering model when creating single traps at different positions. The sound fields simulated by our model are equivalent to the ones obtained using the conventional BEM.

<u>**Solving for the Transducers' Phases for Acoustic 3D Manipulation**</u>

Once we know how to model the extended transmission matrix ($\boldsymbol{E} = \boldsymbol{F} + \boldsymbol{GH}$), the next step is to solve for the transducers' activation $\boldsymbol{\tau}$ that generates levitation traps at target positions in the presence of sound scattering objects. In this study, we assumed phase-only optimization (i.e., the amplitudes of the transducers are always maximum), and thus the goal of this optimization is to find the optimum phases of the transducers ($\boldsymbol{\varphi} = [\varphi_1, \ldots, \varphi_N]^T$) that maximize trapping stiffnesses $\nabla^2 U$ at every trap position $\boldsymbol{r_j}$.

We considered three different levitation solvers: *BASELINE*, *HEURISTIC*, and *SIMPLIFIED*. The *BASELINE* solver uses stiffness, as a physically accurate and broadly accepted metric for trapping quality but is the slowest. The *HEURISTIC* solver is the fastest but not accurate enough. The *SIMPLIFIED* solver represents our solutions being the most balanced, realizing accurate and fast acoustic manipulation.

***BASELINE* Levitation Solver:** One straightforward approach in this optimization problem is, as proposed in (*4*), to directly maximize trapping stiffnesses $\nabla^2 U(\boldsymbol{r_j})$ at every trap position $\boldsymbol{r_j}$ by using a cost function $O(\boldsymbol{\varphi})$ determined as:

$$O(\boldsymbol{\varphi}) = \sum_{j=1}^{J} \left[-\nabla^2 U(\boldsymbol{r_j}) + w_s \left(\overline{\nabla^2 U(\boldsymbol{r_j})} - \nabla^2 U(\boldsymbol{r_j}) \right)^2 \right]. \tag{13}$$

Here, the bar ($\overline{\ }$) represents the mean value among all the J traps; and w_s is a weight coefficient. We added the second term in this cost function to equalize the qualities (i.e., stiffnesses) of all J traps by minimizing the standard deviation similarly to (*41*). The *BASELINE* solver minimizes this cost function $O(\boldsymbol{\varphi})$ in Eq. 13 using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm (*42*, *43*).

However, as already described in the main text, computing trapping stiffnesses $\nabla^2 U$ is computationally heavy because it requires sampling pressure values at many points (e.g., 55 points in this study, which means $L = 55J$) around each trap. The reason it requires so many points is that the second spatial derivative of U requires up to third derivatives of pressure values at the trap position as:

$$\frac{\partial^2 U}{\partial a^2} = 2K_1 \left(\frac{\partial p}{\partial a} \cdot \frac{\partial p}{\partial a} + p \cdot \frac{\partial^2 p}{\partial a^2} \right) - 2K_2 \sum_{b}^{x,y,z} \left(\frac{\partial^2 p}{\partial a \partial b} \cdot \frac{\partial^2 p}{\partial a \partial b} + \frac{\partial p}{\partial b} \cdot \frac{\partial^3 p}{\partial a^2 \partial b} \right), a \in \{x, y, z\}. \tag{14}$$

Here, a represents x, y, or z; and the dot operator ($\cdot$) is defined as: $p_f \cdot p_g = \mathcal{R}e[p_f]\mathcal{R}e[p_g] + \mathcal{I}m[p_f]\mathcal{I}m[p_g]$. To numerically obtain the derivatives in Eq. 14, this metric requires sampling pressure values at many points. In this study, we used the second-order centered difference approximation to compute these derivatives for accuracy because this metric needs to serve as our baseline. Fig. S2a shows how we sampled the pressure values at points around the trap in an ab-plane, where $ab \in \{xy, yz, zx\}$. Note here that p_{10} in the xy-plane is duplicated in the other two planes; and p_9 and p_{11} in the xy-, yz-, and zx-planes are respectively identical to p_6 and p_{14} in the yz-, zx-, and xy-planes. This means, in this study, we used 55 points in total per trap (i.e., 21 for each of the xy-, yz- and zx-planes excluding the $2 + 2 \times 3 = 8$ duplicated points).

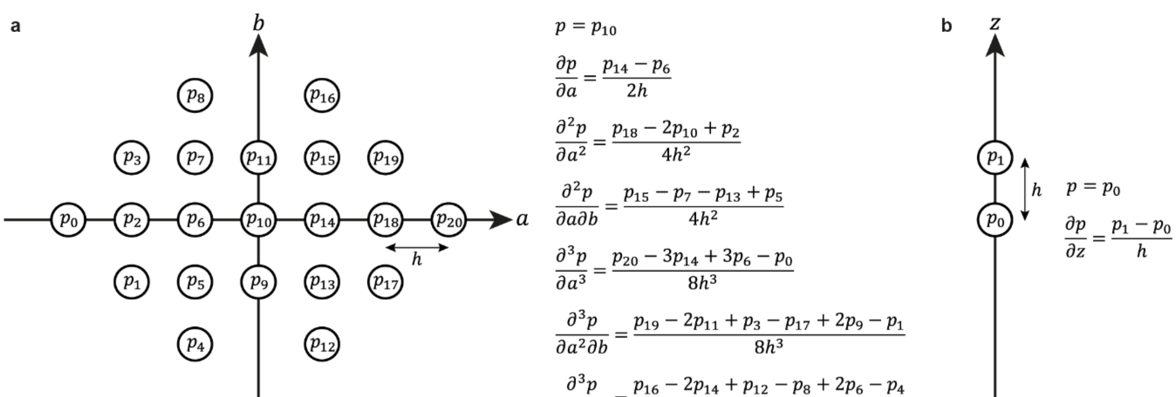

Fig. S2. Sampling points required to compute the metrics. (a) Sampling points in an ab-plane to calculate the trapping stiffness, where $ab \in \{xy, yz, zx\}$. We used 55 points in total per trap (i.e., 21 for each of the xy-, yz- and zx-planes excluding the 8 duplicated points). **(b)** Our proposed simplified metric requires sampling only two points per trap.

***HEURISTIC* Levitation Solver:** In order to simplify this optimization problem, we adapted the heuristic approach proposed in (*6*) for the top-bottom levitation set-ups. This approach uses two points of interest per trap (i.e., $L = 2J$), $\lambda/4$ above and $\lambda/4$

below the position where the trap needs to be located, with a π radian offset in the target phases. By simply back-propagating those points with the conjugate transpose of the transmission matrix $\boldsymbol{E}^*$ and then constraining the transducers' amplitudes to their maxima, we can calculate the transducer phases $\boldsymbol{\varphi}$ without any iterations (i.e., $K = 1$). Although this *HEURISTIC* approach is the simplest and would work well in a single-point manipulation, destructive interference between traps is likely to occur in multi-point manipulation (*15*).

This *HEURISTIC* approach would still work even if the solver used slightly different positions for the two control points, which are at the trap position $\boldsymbol{r_j} = (x_j, y_j, z_j)$ and the position slightly above it $(x_j, y_j, z_j + h)$. This modified version of the *HEURISTIC* levitation solver is also used to obtain initial guesses for the *BASELINE* and *SIMPLIRIFED* solvers (as explained in *Convergence and Initialization*).

SIMPLIFIED **Levitation Solver:** This *SIMPLIFIED* solver uses our proposed simplified Gor'kov potential U' at each trap position as our target cost function, instead of directly using trapping stiffnesses $\nabla^2 U(\boldsymbol{r_j})$. As mentioned in the main text, we determined $U'(\boldsymbol{r_j})$ as:

$$U'(\boldsymbol{r_j}) = K_1 |p|^2 - K_2 \left|\frac{\partial p}{\partial z}\right|^2 ;$$
$$K_1 = \frac{1}{4} V \left(\frac{1}{c_0^2 \rho_0} - \frac{1}{c_p^2 \rho_p}\right);\ K_2 = \frac{3}{4} V \left(\frac{\rho_p - \rho_0}{\omega \rho_0 (\rho_0 + 2\rho_p)}\right). \tag{15}$$

Here, V represents the volume of the levitated particle; ω represents the angular frequency; c and ρ represent the speed of sound and the density, and the subscripts 0 and p refer to the host medium (i.e., air) and the particle material, respectively. The important point here is that $U'(\boldsymbol{r_j})$ can be computed by sampling pressure values at only two points around each trap (i.e., $L = 2J$, see Fig S2b), located at the trap position $\boldsymbol{r_j} = (x_j, y_j, z_j)$ and the position slightly above it $(x_j, y_j, z_j + h)$, in order to numerically compute the derivative along the z-axis (e.g., we used $h = \lambda/32$ in this study). Note here that adding the derivatives along the x- and y-axes (i.e., using the original Gor'kov potential shown in Eq. 1) requires sampling pressure values at four points around each trap (i.e., $L = 4J$). Our simplified metric allows about twice faster update rates when compared to using the original Gor'kov potential (described in *Computational Performance*), but (slower) solutions using the original Gor'kov potential would require minimal changes.

The derivative of $U'(\boldsymbol{r_j})$ with respect to the phase of each transducer φ_n can be computed as:

$$\frac{\partial U'(\boldsymbol{r_j})}{\partial \varphi_n} = 2K_1(\mathcal{Im}[p]\mathcal{Re}[p_n] - \mathcal{Re}[p]\mathcal{Im}[p_n]) - 2K_1\left(\mathcal{Im}\left[\frac{\partial p}{\partial z}\right] \cdot \mathcal{Re}\left[\frac{\partial p_n}{\partial z}\right] - \mathcal{Re}\left[\frac{\partial p}{\partial z}\right] \cdot \mathcal{Im}\left[\frac{\partial p_n}{\partial z}\right]\right). \tag{16}$$

Here, $\mathcal{Re}[\,]$ and $\mathcal{Im}[\,]$ represent real and imaginary parts, and p_n represents a complex pressure value at the j-th trap position created by a single transducer n.

Due to the negative correlation between $\nabla^2 U(\boldsymbol{r_j})$ and $U'(\boldsymbol{r_j})$ (see Fig. 2b and explanation in section *Metric Validity*), we can obtain our cost function $O(\boldsymbol{\varphi})$ to maximize the trapping stiffnesses as:

$$O(\boldsymbol{\varphi}) = \sum_{j=1}^{J} \left[U'(\boldsymbol{r_j}) + w_s\left(\overline{U'(\boldsymbol{r_j})} - U'(\boldsymbol{r_j})\right)^2\right]. \tag{17}$$

The weight coefficient w_s was fixed to 0.0001 in this study. The gradient of this cost function $\nabla O(\boldsymbol{\varphi})$ can be computed as:

$$\frac{\partial O(\boldsymbol{\varphi})}{\partial \varphi_n} = \sum_{j=1}^{J} \left[\frac{\partial U'(\boldsymbol{r_j})}{\partial \varphi_n} + 2w_s\left(\overline{U'(\boldsymbol{r_j})} - U'(\boldsymbol{r_j})\right)\left(\frac{\overline{\partial U'(\boldsymbol{r_j})}}{\partial \varphi_n} - \frac{\partial U'(\boldsymbol{r_j})}{\partial \varphi_n}\right)\right]. \tag{18}$$

Again, computing this gradient requires sampling pressure values at only two points per trap, allowing high-speed computation.

Although any optimization algorithm, such as BFGS, can be used to minimize this cost function $O(\boldsymbol{\varphi})$, we decided to use gradient descent because it is suitable for parallel computation. For further simplicity, we set the step size of the gradient descent algorithm to $-1/\|\nabla O(\boldsymbol{\varphi})\|_2$, which can be determined without using any line searching algorithm. For all evaluations in this study, we set the number of iterations $K = 100$, based on the evaluation in section *Convergence and Initialization*.

Evaluation of Our Acoustic Holographic Technique

In this section, we describe how we evaluated our acoustic holographic technique. In the evaluations, we used four 3D models, *flat*, *smooth*, *bricks*, and *bunny*. We used a polygon mesh processing library (*44*) to uniformly re-mesh the 3D models so that the maximum length of the mesh elements l_{max} is always less than λ, $\lambda/2$, $\lambda/4$, or $\lambda/6$, as shown in Fig. S3. The program detects edges with dihedral angles larger than certain degrees as object features and reserves those features while re-meshing. In most of the evaluations, we used the models with $l_{max} = \lambda/2$, as it is the best-balanced mesh size between speed and accuracy (detailed in *Mesh-size Dependency of the Trap Quality*).

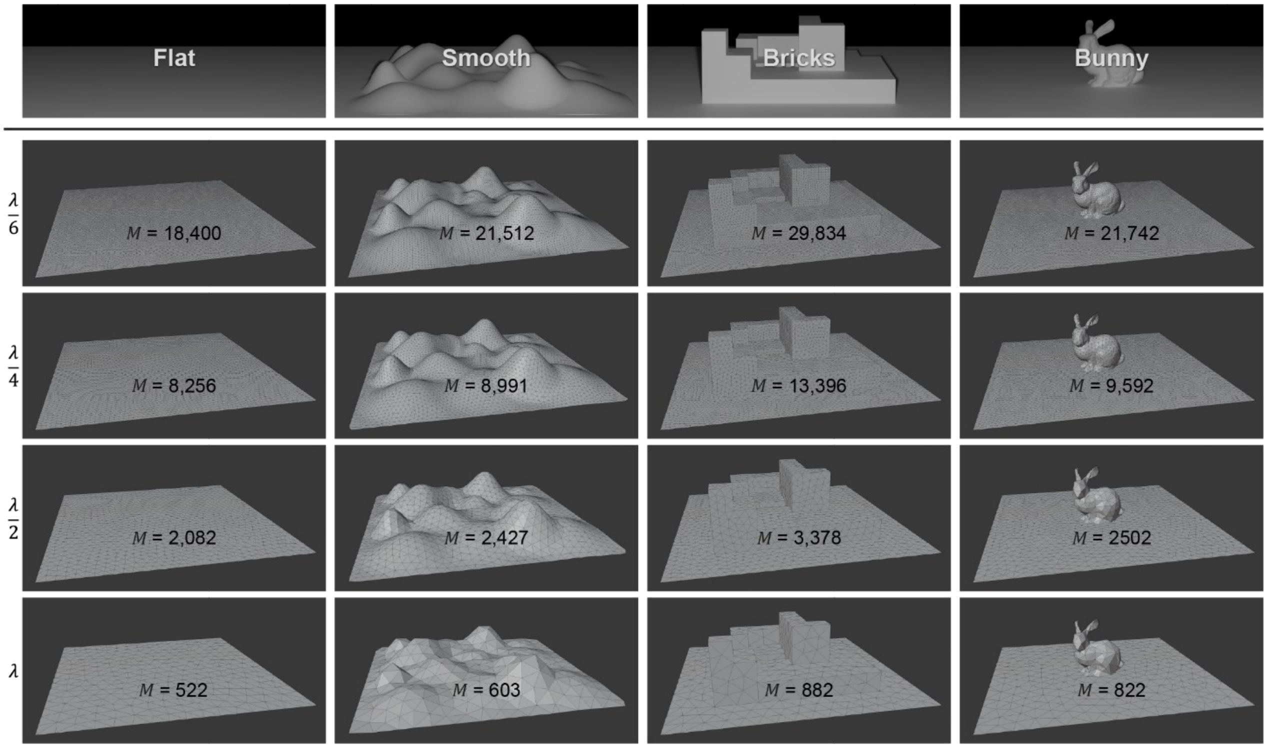

Fig. S3. 3D models used for the evaluations: flat, smooth, bricks, and bunny. The 3D models were re-meshed to have a maximum length of λ, $\lambda/2$, $\lambda/4$, or $\lambda/6$.

Metric Validity: As described earlier, our *SIMPLIFIED* levitation solver uses the simplified Gor'kov potential $U'(\boldsymbol{r_j})$ of Eq. 2 to evaluate the quality of traps, instead of using the trapping stiffness $\nabla^2 U(\boldsymbol{r_j})$. To justify our choice of the metric, we evaluated the correlation between $U'(\boldsymbol{r_j})$ and $\nabla^2 U(\boldsymbol{r_j})$. In this evaluation, the sound-scattering objects with $l_{max} = \lambda/2$ (see Fig. S3) were placed at the origin $(x, y, z) = (0, 0, 0)$, and the PAT was arranged at 12 cm above the objects. We created single traps at 2,000 random arrangements for each of the four objects (i.e., so 8,000 samples in total). Here, the x and y coordinates of the trap positions ranged from -5 to 5 cm, and z was set from 2 to 9 cm. The trap positions that were too close to the objects (i.e., the distance less than 2λ) were excluded. We used the *BASELINE* solver to create the traps and computed $U'(\boldsymbol{r_j})$ and $\nabla^2 U(\boldsymbol{r_j})$ to plot them together (see Fig. 2b). The data obtained can be linearly fit as: $U'(\boldsymbol{r_j}) = b_1 \nabla^2 U(\boldsymbol{r_j}) + b_2$ ($b_1 = -7.23 \times 10^{-7}$ and $b_2 = -1.69 \times 10^{-8}$), with the square of the correlation $R^2 = 0.940$. This correlation indicates that minimizing $U'(\boldsymbol{r_j})$ would result in maximizing the trapping stiffness $\nabla^2 U(\boldsymbol{r_j})$.

Although we confirmed that our simplified Gor'kov potential $U'(\boldsymbol{r_j})$ can be used in our set-ups (i.e., the top array with arbitrary objects), this does not necessarily apply to all experimental set-ups. Here, we demonstrate how our technique can be adjusted to three other PAT set-ups: the top-bottom, V-shape, and single-sided without any reflector. Note here that we assumed using the same 16×16 PAT, but the top-bottom and V-shape ones use two PATs. First, we can use the same simplification (i.e., Eq. 2) for the top-bottom set-up because sound waves propagating in $+z$ and $-z$ directions from the top and bottom arrays can create vertical standing-wave-like traps (see Fig. S4a). In the V-shape set-up with an angle between PATs ($\phi = 90°$), the propagation directions of the two PATs are $(\sin\phi/2, 0, \cos\phi/2)$ and $(-\sin\phi/2, 0, \cos\phi/2)$, respectively. Therefore, thanks to the waves propagating in opposite directions along the x-axis, the following metric enables the creation of strong levitation traps (see Fig. S4b):

$$U'(\boldsymbol{r_j}) = K_1|p|^2 - K_2\left|\frac{\partial p}{\partial x}\right|^2. \tag{19}$$

Note here that the constants K_1 and K_2 are determined by the physical properties of particles and air (see Eq. 15). The single-sided set-up without any reflector is the most challenging of the three due to the absence of the sound wave propagating in the opposite direction. However, we can still create a vortex trap (see Fig. S4c), which is very similar to that already demonstrated in (*4*), by using the following metric:

$$U'(\boldsymbol{r_j}) = K_1|p|^2 - K_2\left(\left|\frac{\partial p}{\partial x}\right|^2 + \left|\frac{\partial p}{\partial y}\right|^2\right). \tag{20}$$

Our two-step scattering model works in any levitation set-up. Thus, by combining it with the levitation solver using the proper metrics, we can create levitation traps with the top-bottom, V-shape, and single-sided set-ups, even in the presence of sound-scattering objects (i.e., the sphere with a radius of 3 cm; see Figs. S4d, S4e, S4f).

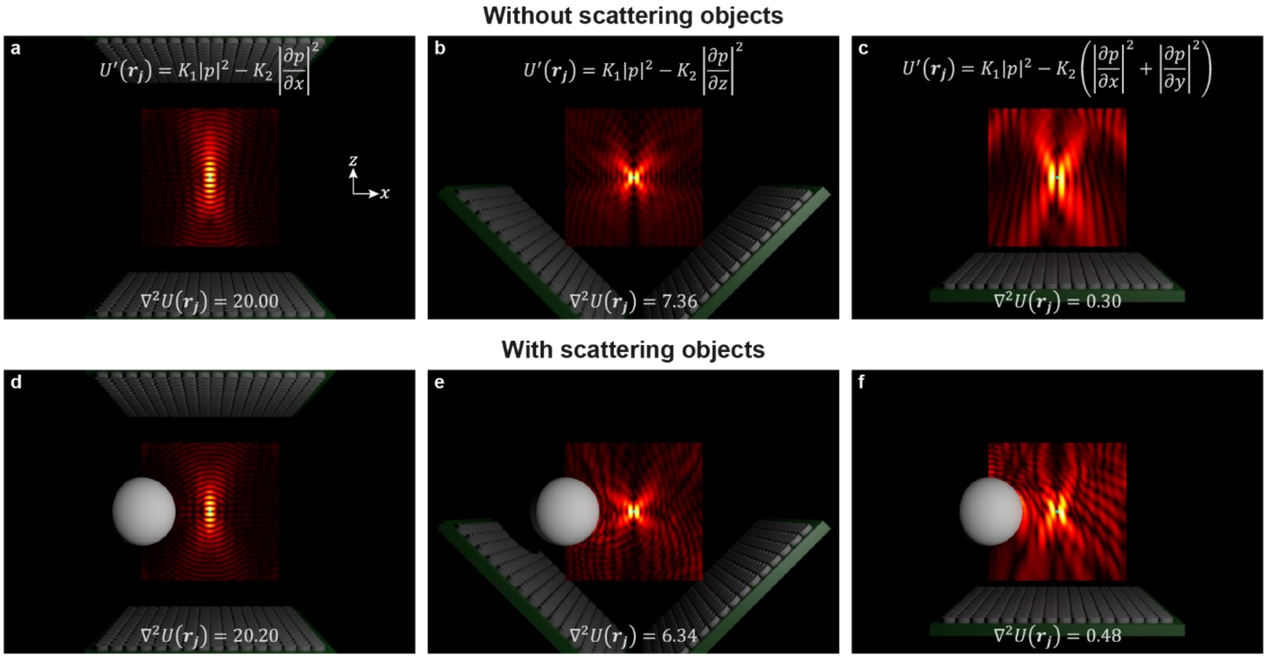

Fig. S4. Other possible metrics for different set-ups. Creation of levitation traps with **(a)** top-bottom, **(b)** V-shape, and **(c)** single-sided set-ups. **(d, e, f)** We can create traps with these set-ups even in the presence of sound-scattering objects (i.e., the sphere with a radius of 3 cm). The trapping stiffness $\nabla^2 U(\boldsymbol{r}_j)$ in each case is shown as a reference.

Distortion and Correction of Sound Fields: To show how sound fields are distorted by sound-scattering objects and how they are corrected by our two-step scattering model, we attempted to create four traps without (*assuming-flat*) and with (*ours*) our model and simulated the generated sound fields. In this evaluation, we used different two 3D models (i.e., *smooth*, *bricks*) in Fig. S3. As the *assuming-flat* model, we used the method of images (*20*). This method can compute sound waves scattering from a flat reflector, by assuming them as the waves emitted by virtual sound sources located at the mirrored positions of the actual sources (i.e., transducers). Therefore, these *assuming-flat* simulations do not account for sound scattering from the objects (i.e., assuming there was only a flat reflector), and thus the generated sound fields can be distorted due to ignoring the presence of the objects. As *ours*, we used our two-step scattering model and compared the results with the *assuming-flat* model (Figs. S5a, S5b). Then, as in the surface-scanning-based display application (Fig. 4d), we horizontally rotated the trap positions and plotted the trapping stiffnesses $\nabla^2 U(\boldsymbol{r}_j)$ at four trap points according to the rotation angle (Figs. S5c, S5d).

Fig. S5 shows that the sound fields are distorted a lot by both of the objects (e.g., the mean trapping stiffnesses decrease 77% and 75% on average, respectively). The *bricks* object is more challenging as it has a non-smooth surface. The minimum trapping stiffness with *bricks* using the *assuming-flat* model becomes even negative (Fig. S5d), suggesting at least one of the four traps is not able to levitate a particle (e.g., the bottom-right trap in the *assuming-flat* image of Fig. S5b). On the other hand, our two-step scattering model can correct such distortion and improve the trapping stiffness by accounting for the sound scattering from the objects.

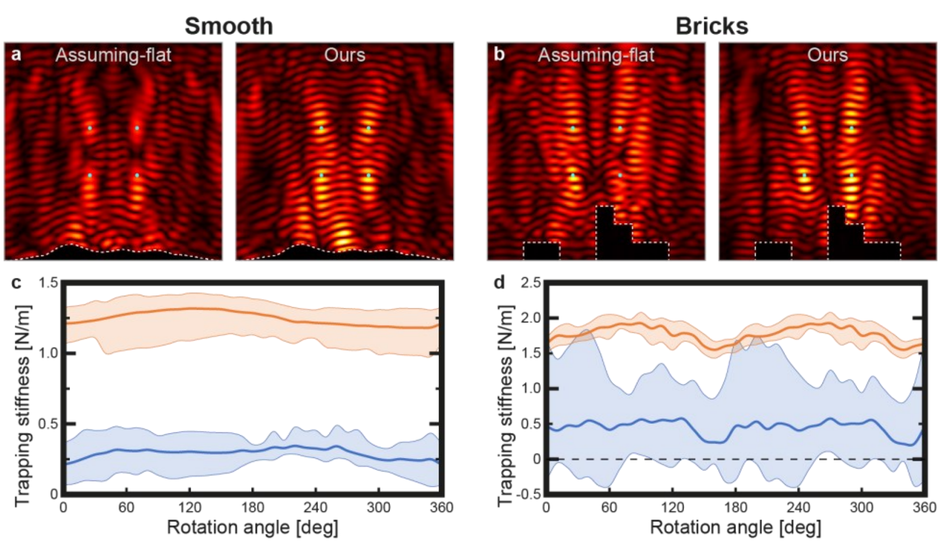

Fig. S5. Distortion and correction of the sound fields. (a, b) Sound-field simulations when creating four traps without (*assuming-flat*) and with (*ours*) our two-step scattering model, with different 3D models (i.e., *smooth*, *bricks*). **(c, d)** Plots showing the mean trapping stiffness of the four traps when the trap positions horizontally rotate. The shaded areas represent minimum and maximum trapping stiffnesses in each case.

Computational Performance: Next, we evaluated the computational performance of our technique using a consumer-grade laptop PC (Intel Core i7-9750H CPU at 2.60 GHz) with a single GPU (NVIDIA GeForce RTX 2080). We used C++ and OpenCL for a parallelized implementation of our method. The positions of traps and mesh elements were randomly generated to be tested as the computational time does not depend on them. We tested 100 times for each combination of the numbers of traps $J = \{1, 2, 4, 8, 16\}$ and mesh elements $M = \{1{,}000, 2{,}000, 4{,}000, 8{,}000, 16{,}000, 32{,}000\}$, and reported the average of the computational times. Note here that in our implementation, the maximum number of frames (i.e., transducers' activation) that the GPU can compute at the same time depends on the number of workgroup size of the GPU (i.e., $N_w = 1{,}024$ in this case) and the number of points of interest required to compute each frame (i.e., $L = 2J$ in our solver), determined as: $N_w/2J$. This indicates the importance of choosing a metric with a small L as it directly relates to the available update rates, for example, using our simplified metric ($L = 2J$) enables the solver to compute about twice faster as using the original Gor'kov potential ($L = 4J$).

Figure 2c summarizes the total computational performance of our technique (i.e., the combination of our model and solver after the pre-computation), for given numbers of transducers ($N = 256$) and iterations for the solver ($K = 100$). Additionally, we tested how fast our scattering model can compute alone to show the breakdown of the computational times (see Fig. S6a). In these plots, the solid lines represent the computational time for only the model, and the dashed lines represent the total computational time (i.e., the same plots as in Fig. 2c). These plots indicate that the solving process becomes more dominant when the number of traps J is higher. This is more notable when the number of iterations K is higher (see Fig. S6b). The numbers of transducers N and traps J are determined by the hardware and applications, respectively, and thus cannot be changed. To reduce the total computational time while keeping sufficient accuracy, the numbers of mesh elements M and iterations K are keys to balancing between speed and accuracy, and we explore these next.

In these performance evaluations, we excluded the set-up-related part (i.e., pre-computation for the matrix $\boldsymbol{H}$) as our main focus is on the ability of our method to retain real-time high-computing rates for applications. Unlike the application-related part, the computational time for the set-up-related part does not depend only on N, L, and M but also on the object geometry. That is, even when two objects have the same number of mesh elements M, the computational times for these objects could differ (e.g., the *flat* reflector is easy to be solved). As references, the pre-computation for the 3D models in Fig. S3 with $l_{max} = \lambda/2$ takes about 9 s for *flat*, 12 s for *smooth*, 21 s for *bricks*, and 17 s for *bunny*, using a naïve CPU implementation.

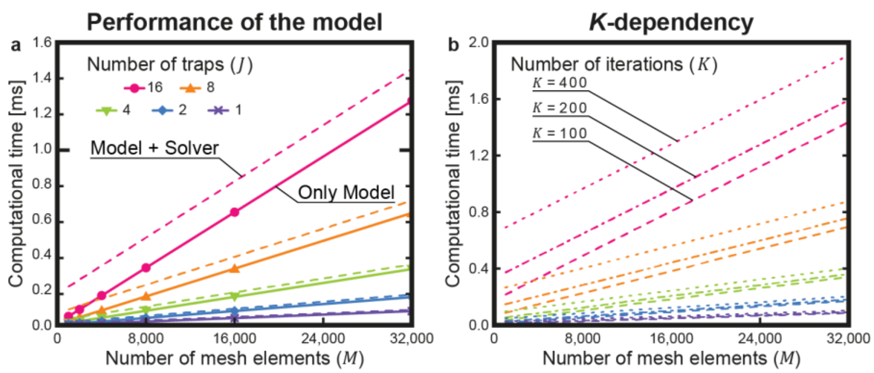

Fig. S6. Computational performance in detail. (a) Computational performance of only the scattering model (solid lines). The dashed lines represent the total computational time (i.e., model + solver excluding the pre-computation) when the number of iterations $K = 100$. **(b)** Total computational performance of the technique with different K.

Mesh-size Dependency of the Trap Quality: As shown in Fig. 2c, the number of mesh elements M is an important parameter that highly affects the computational speed in our technique. The total number of mesh elements depends on the 3D models' mesh resolutions (i.e., the size of the elements), which also influences the accuracy of BEM. In general scattering problems using BEM, six boundary elements per wavelength are usually required for accurate scattering simulations (*45*). However, the purpose of this work is to solve for transducer phases that provide sufficient trapping stiffness, not to accurately simulate sound fields; therefore, such high degrees of freedom per wavelength might not be necessary for our scattering model.

To find the best-balanced size for the mesh elements, we evaluated the mesh-size dependency of the trap quality (i.e., stiffness) using the 3D models in Fig. S3 with different maximum lengths of the mesh elements $l_{max} = \{\lambda, \lambda/2, \lambda/4, \lambda/6\}$. In this evaluation, we created single traps using the *BASELINE* solver at the same trap positions used in the *Metric Validity* test and then simulated the trapping stiffness $\nabla^2(\boldsymbol{r}_j)$ using the finest meshes (i.e., $\lambda/6$). Figure S7 summarizes the mean stiffnesses, showing that the use of $l_{max} = \lambda$ is insufficient for our two-step scattering model, failing to provide enough stiffness (e.g., especially for *smooth* and *bricks*) compared to the subwavelength maximum element sizes. Considering the balance between speed and accuracy, we decided to use $l_{max} = \lambda/2$ in our solver for the rest of the evaluations.

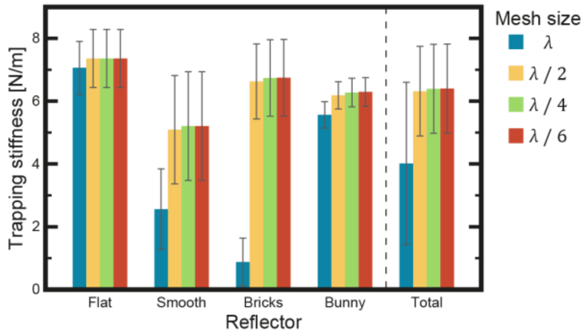

Fig. S7. Mesh-size dependency of the trap quality in our acoustic holographic technique. When the maximum length of the mesh elements $l_{max} = \lambda$, the model fails to provide enough stiffnesses at the trap positions.

Convergence and Initialization: We now show how our *SIMPLIFIED* levitation solver performs on multi-point levitation (i.e., number of traps $J = \{1, 2, 4, 8, 16\}$) in the presence of the four scattering objects used in the previous evaluations (see Fig. S3). We used 1,000 random combinations of trap positions per condition. To avoid cases where traps were too close to each other, we set the minimum distance between the traps to 2λ. Figure S8a shows the average stiffnesses and their standard deviations with different numbers of traps J, with $K = \{10, 20, 40, 80, 100, 200, 400, 800\}$, showing the increase of stiffness along with iterations, when transducer phases were randomly initialized. Even with the highest number of traps (i.e., $J = 16$), we can achieve positive stiffnesses, required for trapping particles, after several iterations.

Figure S8b shows the results when we used the phases obtained using the modified *HEURISTIC* solver instead of random initial phases. The plots demonstrate that the use of such *HEURISTIC* initial guesses reduces the required number of iterations K in the *SIMPLIFIED* solver. Note here that even though the *HEURISTIC* solver already provides comparatively high mean stiffnesses without iterations (i.e., $K = 1$), the iterations are still required to reduce the standard deviation. This is because, in multi-point acoustic levitation, weak traps could fail to hold particles in mid-air (*15*), and the objective is to generate equally strong traps (see more discussion in the next section). The advantage of using the modified version of the *HEURISTIC* solver is that it uses pressure values at exactly the same points with the *SIMPLIFIED* solver (i.e., at the trap position (x_j, y_j, z_j) and the position slightly above it $(x_j, y_j, z_j + h)$) so that we can use the same transmission matrix $\boldsymbol{E}$ for both these initial and iterative steps, without any additional modeling process required. Following these, this *HEURISTIC* initialization and $K = 100$ were used in all the applications and for the rest of the evaluations.

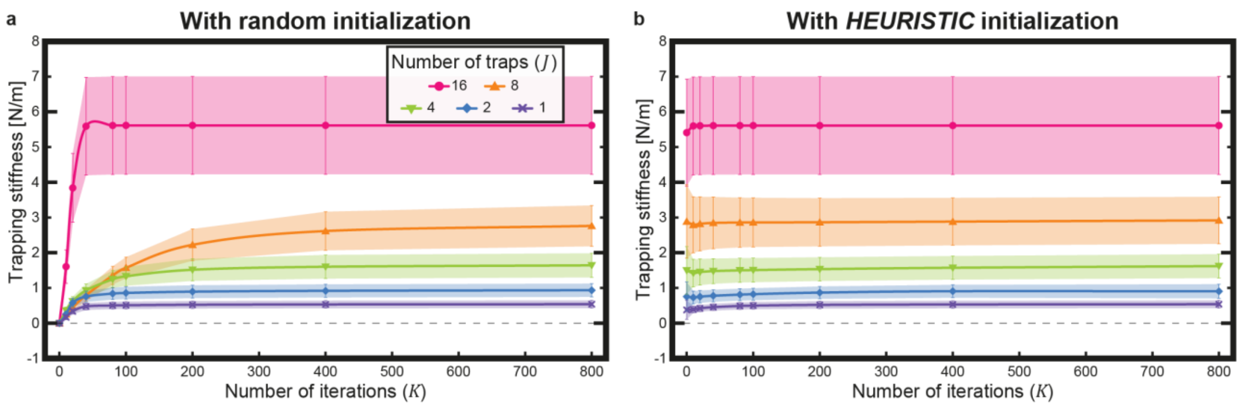

Fig. S8. Convergence of the simplified levitation solver. (a) The trap quality of the *SIMPLIFIED* solver improves depending on the number of iterations K. **(b)** The use of the modified *HEURISTIC* solver as initial guesses reduces the number of iterations required to be converged. The error bars represent standard deviations.

Comparison between the Solvers: In this study, we considered using three solvers: *BASELINE*, *HEURISTIC*, and *SIMPLIFIED*, with our two-step scattering model. Here, we compare these three solvers to demonstrate that only the *SIMPLIFIED* solver provides both high computational speed and trap quality. Similar to the previous evaluation, we used 1,000 random combinations of trap positions per condition (i.e., four scattering objects with the different numbers of traps $J = \{1, 2, 4, 8, 16\}$). The numbers of transducers ($N = 256$) and iterations ($K = 100$) were fixed.

Figure S9a shows the average trapping stiffnesses and their standard deviations obtained by the different solvers. The mean values indicate that *BASELINE* overall is slightly better than *HEURISTIC* and that the performance of *SIMPLIFIED* tends to be between these two. We also confirmed this relationship statistically using the statistics software (i.e., IBM SPSS Statistics 25), as shown in Fig. S9a. The plots also show that *SIMPLIFIED* provides the smallest standard deviations of the solvers. Providing small standard deviations is important in multi-point acoustic levitation to avoid weak traps and realize stable particle manipulation (*15*).

To highlight this point, we performed the same evaluation but focused on the weakest traps of the J traps (see Fig. S9b). The plots indicate that the difference between *HEURISTIC* and the other two becomes more apparent, and *HEURISTIC* likely fails to create traps when the number of traps is large (i.e., negative stiffness with $J = 16$). This is why *HEURISTIC* is not enough even though it offers the fastest computational performance. Figure S9b also shows that *SIMPLIFIED* performs slightly better than even *BASELINE* in terms of the minimum stiffnesses, indicating that *SIMPLIFIED* is more suitable to uniformly provide sufficient stiffness for all the traps in multi-point levitation.

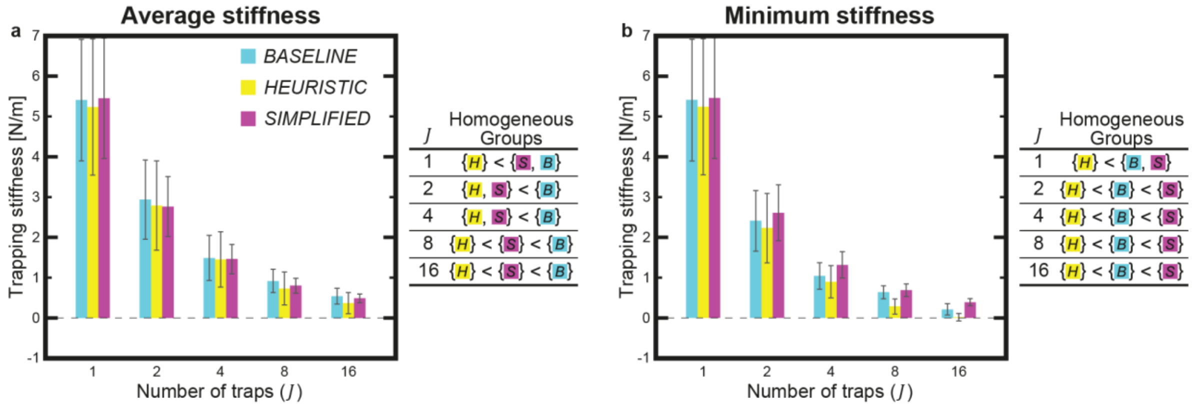

Fig. S9. Comparison of the trap quality between the different solvers. Average stiffnesses **(a)** and minimum stiffnesses **(b)** obtained by the three different solvers. The bars represent standard deviations. The symbol '<' indicates there is a significant difference between the homogeneous groups represented by the symbol '{ }'.

Manipulation Capability:

In this section, we discuss the manipulation capabilities enabled by our technique.

Moving Sound-scattering Objects: In our scattering model, the mesh models remain static over time. This assumption allows us to pre-compute the scattering model (i.e., the matrix $\boldsymbol{H}$). In other words, dealing with dynamic scattering objects is challenging in our acoustic holographic technique. If we know ahead of time the nature of the dynamic evolution of the sound scattering object, different $\boldsymbol{H}$ matrices can be pre-computed, and the other two matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ can be computed in real time. If the sound-scattering object changes in a manner that cannot be predicted ahead of time, we need to repeatedly solve linear equations $\boldsymbol{A}\boldsymbol{p}^{(n)} = \boldsymbol{b}^{(n)}$, where $\boldsymbol{A}$ is an $M \times M$ matrix, for every N transducer to compute $\boldsymbol{H}$ in real time. Note here that, as shown in Eq. 8, the matrix $\boldsymbol{A}$ depends only on the geometry of the scattering objects and not on the positions of the transducers or traps.

One common scenario is where the shape of the scattering object is constant but the object's position or transducers' arrangement changes. In such scenarios, we can assume the object is relatively static by assuming instead the positions of the transducers change. Thus, the matrix $\boldsymbol{A}$ is constant even while the actual position of the object is moving. Therefore, once we decompose this matrix (e.g., by using LU decomposition), we can reuse the decomposed matrices to easily solve the linear equations, obtaining different $\boldsymbol{H}$ at high rates during the movement of the object. Figure S10 shows an example of creating a POV image with a scattering object located at different positions. In these three examples, we used the same lower and upper triangular matrices, which were obtained from the decomposition of $\boldsymbol{A}$, to accelerate the computation of $\boldsymbol{H}$.

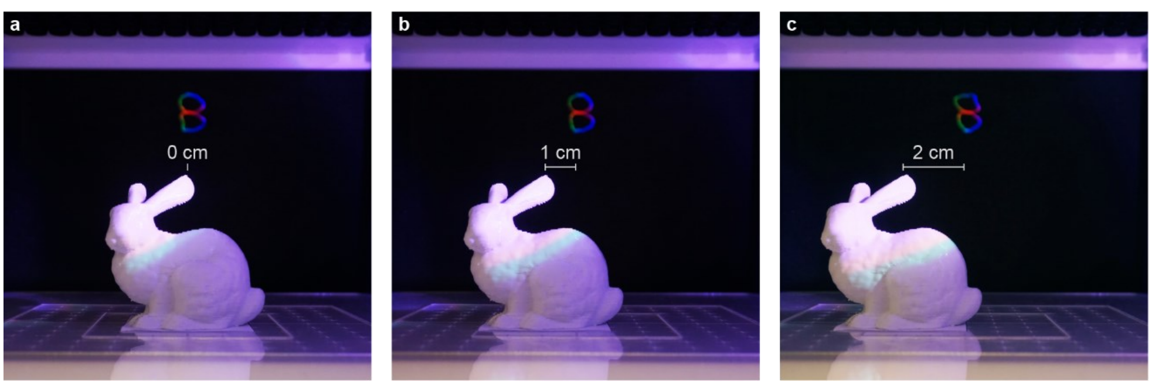

Fig. S10. Creation of POV image with a scattering object located at different locations. (a, b, c) The center of the 3D-printed bunny was located at 0, 1, and 2 cm from the center of the system, respectively.

Scattering Objects Vicinity: One limitation of our technique is the manipulation of particles near the scattering surfaces. When we try to create a trap near a surface, strong sound reflection from the surface tends to create standing-wave-like sound fields on the surface, resulting in the creation of traps at certain discrete heights (z) from the surfaces (i.e., $z = \lambda/4, 3\lambda/4$). Therefore, it is difficult to manipulate a particle from $z = \lambda/4$ to $z = 3\lambda/4$, or vice versa. To show this limitation, we tried to create a single trap with our solver at certain heights (z) from the flat surface (see Fig. S11a) and plot how far the simulated trap positions (i.e., positions where the Gor'kov potential is minimum) were from the target trap positions, even with the *BASELINE* solver (see Fig. S11b). The plot shows very high position errors within the area around $\lambda/2 < z < 3\lambda/4$, indicating failures to create the trap within this area. This manipulation difficulty near scattering surfaces was also confirmed experimentally. Additional research efforts on both algorithmic and hardware fronts (e.g., transducer arrangement) are required for realizing acoustic holographic systems with this feature.

A practical way to bypass this problem is the use of sound-scattering props (see Fig. S11c). Our two-step scattering model enables us to manipulate a particle along the ramp by creating traps $\lambda/4$ over the ramp surface. Once the particle is high enough from the surface (e.g., $z \geq 3\lambda/4 = 6.49$ mm), we can push the particle off the ramp and manipulate it in 3D without any constraint. We have experimentally confirmed this approach works to pick up particles from the ground.

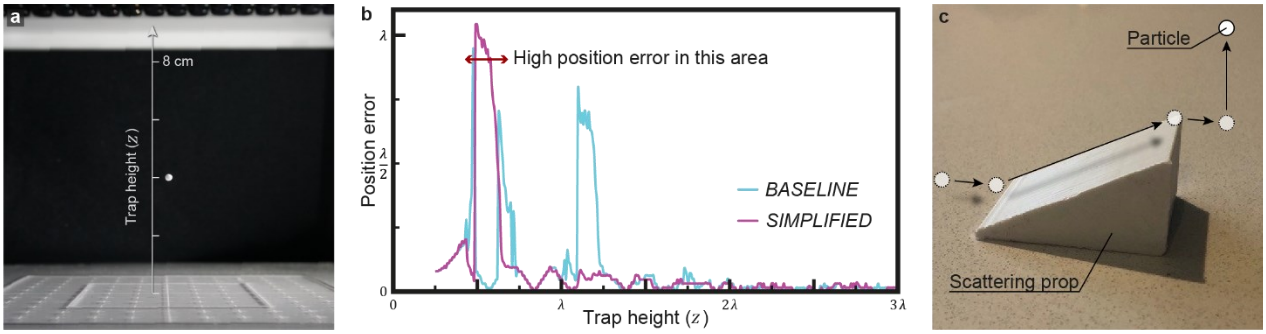

Fig. S11. Limitation on acoustic manipulation near scattering surfaces. (a) Experimental set-up. **(b)** Position error near the flat surface depending on the trap height (z). Here, the wavelength $\lambda = 8.65$ mm in this study. **(c)** Sound-scattering prop enabling to translate particles near the surface.

Handling Liquid Droplets: Consider we manipulate a liquid droplet horizontally as shown in Fig. S12a. In acoustic manipulation of liquid droplets, the ratio of acoustic forces to surface forces for a levitated droplet is described by the acoustic Bond number (*2*, *33*) as: $B_a = 2v_{rms}^2 \rho_0 R_s / \sigma$, where σ is the surface tension of the liquid; R_s is the droplet radius, and v_{rms} is the root mean square of the acoustic velocity of air particles. To avoid atomization of the levitated droplet (i.e., droplet bursting), this acoustic Bond number needs to be between 2.5 and 3.6, as experimentally determined in (*33*). Therefore, it is important to keep the acoustic velocity constant along manipulation paths. In our experiment, we manipulated a liquid droplet horizontally (see Fig. S12a and Movie S2). The fast computational rates of our technique enable us to estimate the acoustic velocity in real time and to adjust the transducers' amplitudes to make the acoustic velocity constant along the manipulation path (see Fig. 9b).

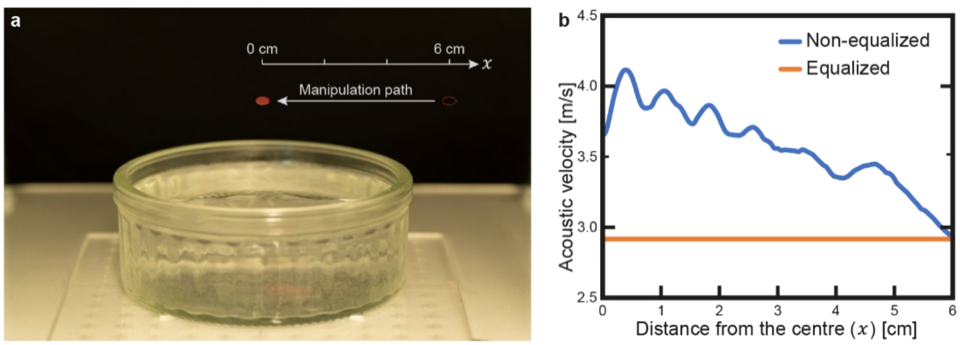

Fig. S12. Equalization of acoustic velocity along levitation paths to avoid atomization of droplets. (a) Definition of the manipulation path of a pain droplet. **(b)** Adjusting the amplitudes of the transducers depending on the estimated acoustic velocity (i.e., the velocity of air particles) allows the handling of liquid droplets without causing atomization.

<u>MR Applications</u>

In this section, we describe how we created the MR applications.

Experimental Set-ups: All our applications used the same levitation set-ups. The applications were created using a single PAT of 16 × 16 transducers, designed as an extension of the Ultraino platform (*6*), modified for faster communication rates as (*15*). The array used Murata MA40S4S transducers (40 kHz, 10.5 mm diameter (~1.2λ), delivering ~8.1 Pa at 1-m distance when driven at 20 Vpp). A Waveshare CoreEP4CE10 Field Programmable Gate Array (FPGA) board was used to receive phase and amplitude updates from the CPU, using a USB FT245 Asynchronous FIFO Interface at 8 Mbyte/sec and allowing more than 10,000 phase and amplitude updates per second. The PAT and a base flat acrylic reflector were aligned on top of each other with an adjustable separation (e.g., fixed to 12 cm in this study). A square part (12 × 12 cm^2) of the flat reflector can be replaced by arbitrary scattering surfaces, such as 3D-printed ones, sets of bricks, and a glass container filled with water. We used a LulzBot mini 3D printer with eSUN PLA+ filament to 3D-print the objects. For the interactive applications (see Movie S4), we used a LeapMotion sensor to detect the user's fingertip positions.

Mid-air Screen: We used the same method described in (*12*) to prepare the mid-air screen for levitation. We first laser-cut light, acoustically transparent fabric (Super Organza) into a square of 3 × 3 cm^2. Four EPS particles were glued on the piece of fabric, acting as anchors to allow 6-degrees-of-freedom manipulation of the fabric. For projection mapping onto this levitated fabric, we used a projector (Texas Instruments, DLP LightCrafter Evaluation Module) with a native resolution of 608 × 684 pixels. We obtained the intrinsic parameters of this projector in advance by using an OpenCV function (*calibrateCamera*) with a checkerboard and a web camera, and then obtained the extrinsic parameters (i.e., positions and orientation, relative to the levitator coordinate) by using the manually collected combinations of trap positions in the levitator coordinate and pixel positions in the projector coordinate. We then used such parameters for our OpenGL cameras (i.e., projection and view matrices) to enable real-time projection mapping (see Fig. 1b).

Point-scanning-based Volumetric Display: In these applications, we used high-intensity full-color LEDs (OptoSupply, OSTCWBTHC1S) to illuminate the levitated EPS particles. The LEDs were directly controlled by the FPGA, which controls the transducers as well so that the illumination colors and the movements of the levitated particles were synchronized. All the scanning paths were generated to be scanned by the particles in the POV time (i.e., 0.1 s). Therefore, we were able to create the volumetric POV images (see Figs. 4a, 4b, and 4c).

Note here that in the point-scanning-based approach, the maximum number of voxels N_v is determined by the update rate of the levitator f_l, the number of traps J, and the POV rate ($f_{POV} = 10$ Hz) as: $N_v = J \cdot f_l / f_{POV}$ (e.g., N_v = 4,000 when $f_l = 10{,}000$ and $J = 4$). Also, there are additional constraints in the voxel arrangement because the paths created by these voxels need to be scanned by single or multiple points. That is, the voxels need to be continuous, and the particle movements along the voxel paths need to be within the system's capabilities (i.e., maximum velocity and acceleration). These constraints make it difficult to create complex volumetric shapes with the point-scanning-based approach.

Surface-scanning-based Volumetric Display: We re-used the same fabric, projector, and calibration scheme used in the mid-air screen application. However, in this application, we used the projector in a high-speed binary mode at 1,440 fps. As shown in Fig. 4d, we placed a mirror in the system to cover the angles, where the projector is not capable of directly projecting onto the fabric (i.e., when the fabric and projection direction become parallel). In other words, we used the mirror as a second projector. We created 144 cross-sectional binary images of a 3D model (i.e., bunny) every 1.25 degrees, mapped those images onto the rotating screen, and encoded them into 24-bit images as in (*46*). Then the system levitated and rotated the fabric at five rotations per second while updating the encoded images at 60 Hz. Our OpenGL-based software can adjust the timing of projecting the cross-sectional images so that it matches the fabric's rotational timing. The software also receives a VSYNC signal to automatically adjust the timing of projecting the cross-sectional images corresponding to the levitator update.

In the surface-based approach, the maximum number of voxels of created images N_v is determined by the update rate of the projector f_p and the number of pixels of projected 2D images N_p as: $N_v = N_p \cdot f_p / f_{POV}$. Thus, ideally, $N_v = 608 \times 684 \times 1{,}440 / 10 \cong 60{,}000{,}000$, which is almost 15,000 times larger than the point-based approach. Although it is not realistic to assume full use of the pixels with a static projector like in our current system, it is possible to increase the usage of the pixels to nearly 100% by utilizing a projection engine with a rotational mirror such as demonstrated in (*39*). Additionally, the voxel arrangement is independent of the content because it is fixed, so the displayed content does not need to account for the levitator's capabilities (i.e., velocity and acceleration), once the levitator is able to rotate the fabric at five rotations per second.

References

1. M. A. B. Andrade, A. Marzo, J. C. Adamowski, Acoustic levitation in mid-air: Recent advances, challenges, and future perspectives. *Appl. Phys. Lett.* **116**, 250501 (2020).
2. D. Foresti, M. Nabavi, M. Klingauf, A. Ferrari, D. Poulikakos, Acoustophoretic contactless transport and handling of matter in air. *Proc. Natl. Acad. Sci. U. S. A.* **110**, 12549–54 (2013).
3. Y. Ochiai, T. Hoshi, J. Rekimoto, Pixie Dust: Graphics Generated by Levitated and Animated Objects in Computational Acoustic-Potential Field. *ACM Trans. Graph.* **33**, 1–13 (2014).
4. A. Marzo, S. A. Seah, B. W. Drinkwater, D. R. Sahoo, B. Long, S. Subramanian, Holographic acoustic elements for manipulation of levitated objects. *Nat. Commun.* **6**, 8661 (2015).
5. K. Melde, A. G. Mark, T. Qiu, P. Fischer, Holograms for acoustics. *Nature*. **537**, 518–522 (2016).
6. A. Marzo, B. W. Drinkwater, Holographic acoustic tweezers. *Proc. Natl. Acad. Sci.*, 201813047 (2018).
7. X. Ding, S. C. S. Lin, B. Kiraly, H. Yue, S. Li, I. K. Chiang, J. Shi, S. J. Benkovic, T. J. Huang, On-chip manipulation of single microparticles, cells, and organisms using surface acoustic waves. *Proc. Natl. Acad. Sci. U. S. A.* **109**, 11105–11109 (2012).
8. T. Vasileiou, D. Foresti, A. Bayram, D. Poulikakos, A. Ferrari, Toward Contactless Biology: Acoustophoretic DNA Transfection. *Sci. Rep.* **6**, 20023 (2016).
9. D. Foresti, K. T. Kroll, R. Amissah, F. Sillani, K. A. Homan, D. Poulikakos, J. A. Lewis, Acoustophoretic printing. *Sci. Adv.* **4**, eaat1659 (2018).
10. T. Omirou, A. Marzo, S. A. Seah, S. Subramanian, LeviPath: Modular Acoustic Levitation for 3D Path Visualisations. *Proc. ACM CHI'15 Conf. Hum. Factors Comput. Syst.* **1**, 309–312 (2015).
11. D. R. Sahoo, N. Takuto, A. Marzo, T. Omirou, M. Asakawa, S. Subramanian, JOLED: A Mid-Air Display Based on Electrostatic Rotation of Levitated Janus Objects. *29th ACM User Interface Softw. Technol. Symp. (UIST '16)*, 437–448 (2016).
12. R. Morales, A. Marzo, S. Subramanian, D. Martínez, LeviProps: Animating Levitated Optimized Fabric Structures using Holographic Acoustic Tweezers. *Proc. 32nd ACM User Interface Softw. Technol. Symp. - UIS* (2019), doi:10.1145/3332165.3347882.
13. R. Hirayama, D. Martinez Plasencia, N. Masuda, S. Subramanian, A volumetric display for visual, tactile and audio presentation using acoustic trapping. *Nature*. **575**, 320–323 (2019).
14. T. Fushimi, A. Marzo, B. W. Drinkwater, T. L. Hill, Acoustophoretic volumetric displays using a fast-moving levitated particle. *Appl. Phys. Lett.* **115**, 064101 (2019).
15. D. M. Plasencia, R. Hirayama, R. Montano-Murillo, S. Subramanian, GS-PAT: High-speed Multi-point Sound-fields for Phased Arrays of Transducers. *ACM Trans. Graph.* **39** (2020), doi:0.1145/3386569.3392492.
16. V. Paneva, A. Fleig, D. M. Plasencia, T. Faulwasser, J. Müller, OptiTrap: Optimal Trap Trajectories for Acoustic Levitation Displays. *ACM Trans. Graph.*, 1–25 (2022).
17. I. Sutherland, The Ultimate Display. *Proc. IFIPS Congr.* **65**, 506–508 (1965).
18. M. A. B. Andrade, N. Perez, F. Buiochi, J. C. Adamowski, Matrix method for acoustic levitation simulation. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*. **58**, 1674–1683 (2011).
19. M. A. B. Andrade, T. S. A. Camargo, A. Marzo, Automatic contactless injection, transportation, merging, and ejection of droplets with a multifocal point acoustic levitator. *Rev. Sci. Instrum.* **89** (2018), doi:10.1063/1.5063715.
20. L. E. Kinsler, A. R. Frey, A. B. Coppens, J. V. Sanders, in *Wiley-VCH* (Princeton University Press, Princeton, 1999; http://www.degruyter.com/view/books/9781400881734/9781400881734-002/9781400881734-002.xml).

21. T. Carter, S. A. Seah, B. Long, B. Drinkwater, S. Subramanian, UltraHaptics: Multi-point mid-air haptic feedback for touch surfaces. *UIST 2013 - Proc. 26th Annu. ACM Symp. User Interface Softw. Technol.*, 505–514 (2013).
22. B. Long, S. A. Seah, T. Carter, S. Subramanian, Rendering volumetric haptic shapes in mid-air using ultrasound. *ACM Trans. Graph.* **33**, 1–10 (2014).
23. P. Glynne-Jones, C. E. M. Démoré, C. Ye, Y. Qiu, S. Cochran, M. Hill, Array-controlled ultrasonic manipulation of particles in planar acoustic resonator. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*. **59**, 1258–1266 (2012).
24. M. A. Norasikin, D. Martinez Plasencia, S. Polychronopoulos, G. Memoli, Y. Tokuda, S. Subramanian, SoundBender: Dynamic Acoustic Control Behind Obstacles. *31st Annu. ACM Symp. User Interface Softw. Technol. - UIST '18*, 247–259 (2018).
25. Y. Liu, *Fast Multipole Boundary Element Method: Theory and Applications in Engineering* (Cambridge University Press, Cambridge, 2009; http://ebooks.cambridge.org/ref/id/CBO9780511605345).
26. S. N. Chandler-Wilde, S. Langdon, in *Unified Transform for Boundary Value Problems: Applications and Advances*, A. S. Fokas, B. Pelloni, Eds. (SIAM, 2014), pp. 181–222.
27. S. Inoue, S. Mogami, T. Ichiyama, A. Noda, Y. Makino, H. Shinoda, Acoustic Macroscopic Rigid Body Levitation by Responsive Boundary Hologram. *arXiv*. **328** (2017), doi:10.1121/1.5087130.
28. J. Greenhall, F. Guevara Vasquez, B. Raeymaekers, Ultrasound directed self-assembly of user-specified patterns of nanoparticles dispersed in a fluid medium. *Appl. Phys. Lett.* **108**, 103103 (2016).
29. M. Prisbrey, J. Greenhall, F. Guevara Vasquez, B. Raeymaekers, Ultrasound directed self-assembly of three-dimensional user-specified patterns of particles in a fluid medium. *J. Appl. Phys.* **121**, 014302 (2017).
30. T. Kozuka, K. Yasui, T. Tuziuti, A. Towata, Y. Iida, Acoustic standing-wave field for manipulation in air. *Jpn. J. Appl. Phys.* **47**, 4336–4338 (2008).
31. H. Bruus, Acoustofluidics 7: The acoustic radiation force on small particles. *Lab Chip*. **12**, 1014–21 (2012).
32. T. Fushimi, B. W. Drinkwater, T. L. Hill, What is the ultimate capability of acoustophoretic volumetric displays? *Appl. Phys. Lett.* **116**, 244101 (2020).
33. C. P. Lee, A. V. Anilkumar, T. G. Wang, Static shape of an acoustically levitated drop with wave-drop interaction. *Phys. Fluids*. **6**, 3554–3566 (1994).
34. A. Watanabe, K. Hasegawa, Y. Abe, Contactless fluid manipulation in air: Droplet coalescence and active mixing by acoustic levitation. *Sci. Rep.* **8**, 1–8 (2018).
35. R. W. Bowen, J. Pola, L. Matin, Visual persistence: Effects of flash luminance, duration and energy. *Vision Res.* **14**, 295–303 (1974).
36. D. E. Smalley, E. Nygaard, K. Squire, J. Van Wagoner, J. Rasmussen, S. Gneiting, K. Qaderi, J. Goodsell, W. Rogers, M. Lindsey, K. Costner, A. Monk, M. Pearson, B. Haymore, J. Peatross, A photophoretic-trap volumetric display. *Nature*. **553**, 486–490 (2018).
37. J. Berthelot, N. Bonod, Free-space micro-graphics with electrically driven levitated light scatterers. *Opt. Lett.* **44**, 1476 (2019).
38. J. Geng, Three-dimensional display technologies. *Adv. Opt. Photonics*. **5**, 456 (2013).
39. G. E. Favalora, J. Napoli, D. M. Hall, R. K. Dorval, M. Giovinco, M. J. Richmond, W. S. Chun, 100 Million-voxel volumetric display. *Proc. SPIE*. **4712**, 300–312 (2002).
40. Y. Saad, M. H. Schultz, GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. Sci. Stat. Comput.* **7**, 856–869 (1986).
41. S. Polychronopoulos, G. Memoli, Acoustic levitation with optimized reflective metamaterials. *Sci. Rep.* **10**, 4254 (2020).
42. Dong C. Liu, Jorge Nocedal, On the limited memory BFGS method for large scale optimization. *Math. Program.* **45**, 503–528 (1989).
43. S. G. Nash, J. Nocedal, A Numerical Study of the Limited Memory BFGS Method and the Truncated-Newton Method for Large Scale Optimization. *SIAM J. Optim.* **1**, 358–372 (1991).
44. D. Sieger, M. Botsch, The polygon mesh processing library (2020), (available at http://www.pmp-library.org).
45. S. Marburg, Six boundary elements per wavelength: Is that enought? *J. Comput. Acoust.* **10**, 25–51 (2002).
46. A. Jones, I. McDowall, H. Yamada, M. Bolas, P. Debevec, Rendering for an interactive 360° light field display. *ACM Trans. Graph.* **26**, 40 (2007).

Acknowledgements

We thank E. Haynes from University College London, who helped with the supplementary videos. **Funding**: This work was supported by EU's H2020 program through their ERC Advanced Grant (No. 787413) and the Royal Academy of Engineering Chairs in Emerging Technology Scheme (CiET1718/14). **Author contributions**: R.H. and S.S. conceived the concept and designed the research. R.H. and G.C developed the mathematical model and solver, with contributions from D.M.P and S.S. R.H. and D.M.P. implemented the algorithms and designed the software. Data analysis was led by R.H., with contributions from all authors. R.H. wrote the paper, with contributions from all authors. **Competing interests**: The authors declare no competing interests. **Data and materials availability**: All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials. Additional data is archived at https://doi.org/10.5281/zenodo.6366502.