Demonstrating Paraflow: Interactive fluid dynamics simulation with real-time visualization for augmented resin 3D printing

GABRIEL LIPKOWITZ, Stanford University, USA

JOSEPH M. DESIMONE, Stanford University, USA

Abstract

While resin 3D printers are seeing growing adoption in both manufacturing and personal fabrication settings, detecting print failures in real time remains challenging. Object-detection neural networks have shown benefits in a variety of extrusion-based 3D printing methods. Here, we extend such work to resin printing using a physics-informed machine learning data generation pipeline. Our approach leverages our models of the fluid dynamics of the printing process at every slice, in order to synthetically generate a library of print defects. We show such an approach is capable of providing data sufficiently resembling real-world failures to fine-tune a pre-trained custom defect detection neural network that can alert users of failure in real-time. Finally, to allow novice users to take advantage of our simulation platform, we integrate our tool into an interactive augmented reality interface, which displays simulation predictions to provide guidance on design and machine parameters prior to printing.

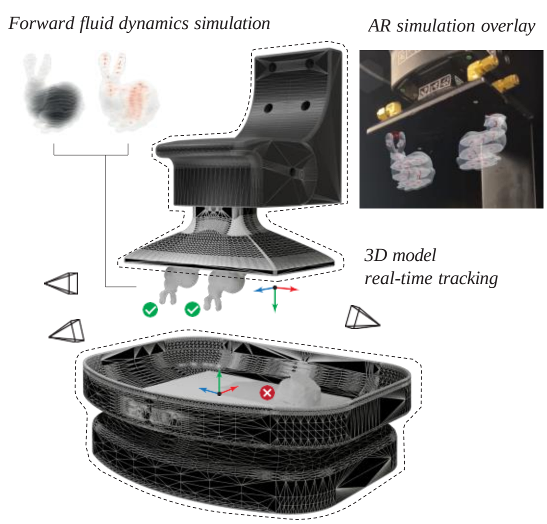


Fig. 1. Summary of our interactive fluid dynamics simulation and visualization tool for augmented resin 3D printing.

1 Introduction and problem formulation

Resin 3D printing is finding increasing adoption in real-world fabrication settings, not only in manufacturing plants but also among interested hobbyists and everyday makers [38]. However, challenges associated with predicting and correcting failures in such systems preclude their more widespread use, particularly among novice users, hindering the promise of decentralizing manufacturing, personalizing fabrication, and democratizing product design. While a variety of tools helping users debug print failures are available for extrusion-based printers, both academic [11, 39] and open-source though online platforms such as *Instructables* and *All3DP*, comparatively fewer are available for resin 3D printers.

In resin-based 3D printing, which has applications in high resolution biomedical models [41], architectural scale models [8], and fashion [35], among other areas, objects are cured through a projection of UV light through a transparent window underneath the resin bath. The separation forces required to subsequently detach the part from the window, or overcome suction forces in the case of continuous liquid interface production (CLIP) [33], are the primary cause of print failure in resin printing. Also known as Stefan adhesion forces, these exert significant stresses on the growing object each layer, and have been studied experimentally in detail by many [24, 12]. If left uncontrolled, these forces can cause detachment of the part from the platform (adhesive failure) or delamination of newly cured layers (cohesive failure) [15]), in either case wasting time and material.

Recently, we have developed a novel simulation pipeline for modeling resin 3D printing that, at every layer during printing, predicts suction forces to anticipate print failure [19, 18]. Here, we leverage such simulation capabilities in a machine learning pipeline for resin 3D printing defect detection, synthetically generating print failures in virtual worlds in order to train a custom defect detection model. To enable particularly new users to use our simulation engine, we wrap our tool in an interactive augmented reality interface, displaying simulation predictions in the context of real-world print failures and the machine environment. In sum, our novel contributions are:

- A physics-informed synthetic data pipeline to detect failures during resin 3D printing; and,
- An augmented reality interface that overlays such simulation data on the real-world printer context.

2 Simulation-guided data generation for real-time print monitoring

We frame the major class of resin printing failure described above – delamination of the object from the build platform – as an object detection task. Our specific aim is to detect the scenario where an object is, or potentially multiple objects are, no longer attached to the platform, and instead has, or have, delaminated in the vat. On such object detection tasks, deep learning-based models have achieved state-of-the-art performance in a variety of domains [20], including in the context of defect detection in additive manufacturing [9, 14, 16]. Other researchers have developed neural networks for the purpose of defect detection in additive manufacturing, but these have almost exclusively been in the context of extrusion printing [4]. Moreover, such defect

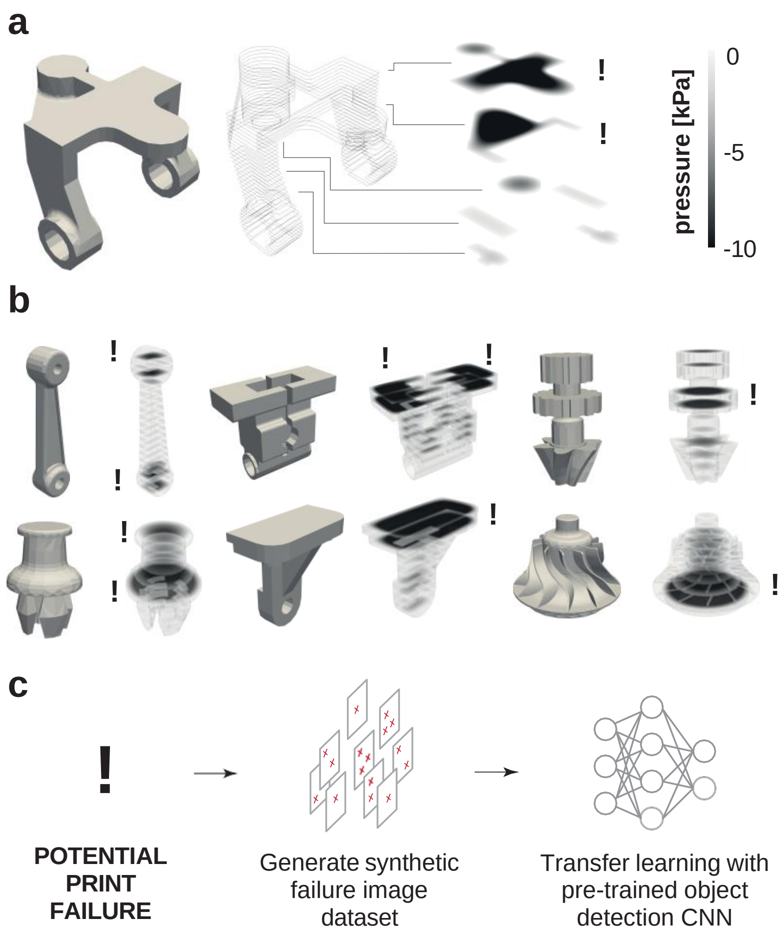


Fig. 2. We predict plausible print failure scenarios with our forward fluid dynamics model (a), illustrated for a range of illustrative mechanical designs (b) using these predictions to generate a synthetic image failure dataset for training a custom object detection model (c).

detection models rely on the collection and manual labeling of real-world data [25, 5]; however, this is tedious and cumbersome, and would be unfeasible for a user initiating a print with an unseen design. Recent work in the machine learning community suggests that simulation engines, akin to that which we develop in this work, can also be effectively leveraged for training neural networks for detection purposes, specifically through the use of synthetic data [22]. Here, the use of computer graphics engines for generating high-quality, perfectly labeled image data has been shown to approach real-world data performance on a variety of computer vision tasks [3, 32, 21]. Drawing from this emerging literature, we explore the possibility of using our simulation engine as a synthetic data generation pipeline. Our overall system architecture is summarized in **Figure 2**, which we explain in more detail below.

2.1 Synthetic data generation pipeline

We outline our data pipeline approach in **Algorithm 1**. In brief, we anticipate likely print failure scenarios with our forward fluid dynamics simulation framework [19, 18], using these scenarios to generate a library of synthetic image failures for training a custom object detection model. This approach is summarized in **Figure 3**. In addition to simulating print failure objects themselves, for environmental context (including distractors and object aggressors), we obtain a

3D scan of the printer environment, along with exact CAD models of the printer build area and models to-be-printed, to render photorealistic images with segmentation masks. To improve detection performance on potentially highly variable real-world data – with multiple camera poses set by the user, under potentially variable lighting conditions, and with diverse failure manifestations – we employ domain randomization, which has been shown to be important for bridging the sim-to-real gap and achieving satisfactory performance on real-world detection [31]. Specifically, we systematically vary aspects of the scene such as lighting, camera pose, and material textures. For implementation, we utilize Blender (version 3.3.0), a popular open-source 3D creation software [2], and its versatile Python API. A sample of synthetic images generated by this pipeline, along with real-world counterparts, is shown in **Figure 4**, for the case of one, two, or three print failures. We use this generated synthetic data for transfer learning with a single pass convolutional neural network, Yolov5 [30], pre-trained on 300,000 open-source data images from the COCO dataset [17]; we train Yolov5s specifically. For all training experiments, we use following hyperparameters: batch size of 8, learning rate of 0.01, train-test split ratio of 90-10, momentum of 0.937. Training was performed on NVIDIA Quadro P5000 GPU.

Algorithm 1: PRINT FAILURE DETECTION DATA PIPELINE

```
Function Generate Synthetic Data(W,M): /* Printer P, 3D models M */
    Initialize W with printer P                         /* synthetic world W */
    Initialize S ← {}                                    /* synthetic images S */
    foreach i = 1, 2, . . . , i_max do                   /* data iteration i */
        foreach m = 1, 2, . . . , m_max do               /* 3D model m */
            Randomly select slice failure s_f      /* slice of suction failure s_f */
            Jitter x, y position of m
            Deform mesh x, y, z dimensions of m
            Randomize position and orientation of m
            Perturb material texture of m
        end
        Randomize scene W lighting
        Randomize camera position and orientation
        Render scene W to images S
    end
    return S
```

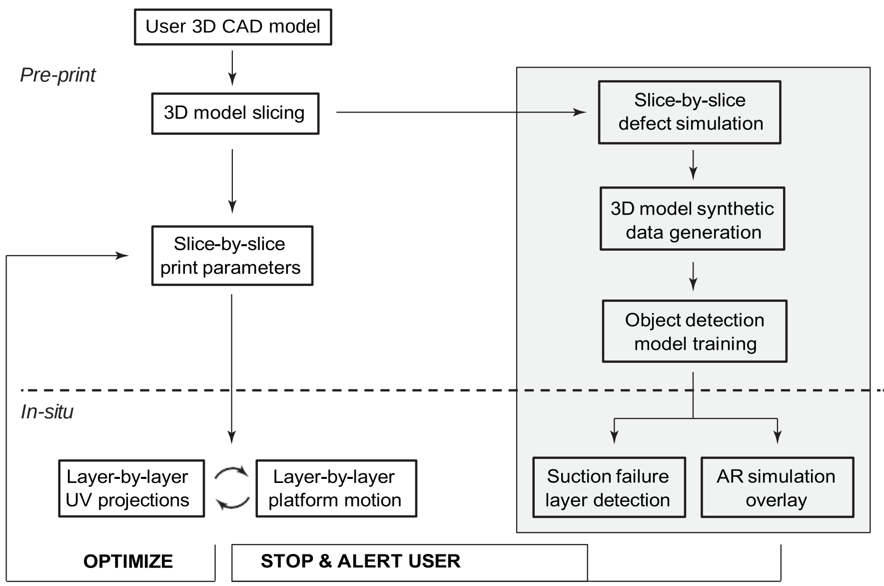


Fig. 3. System architecture for our simulation-guided data generation pipeline to augmented existing 3D printing slicing workflows. To left is depicted a typical 3D printing workflow, and to right our complementary software tool.

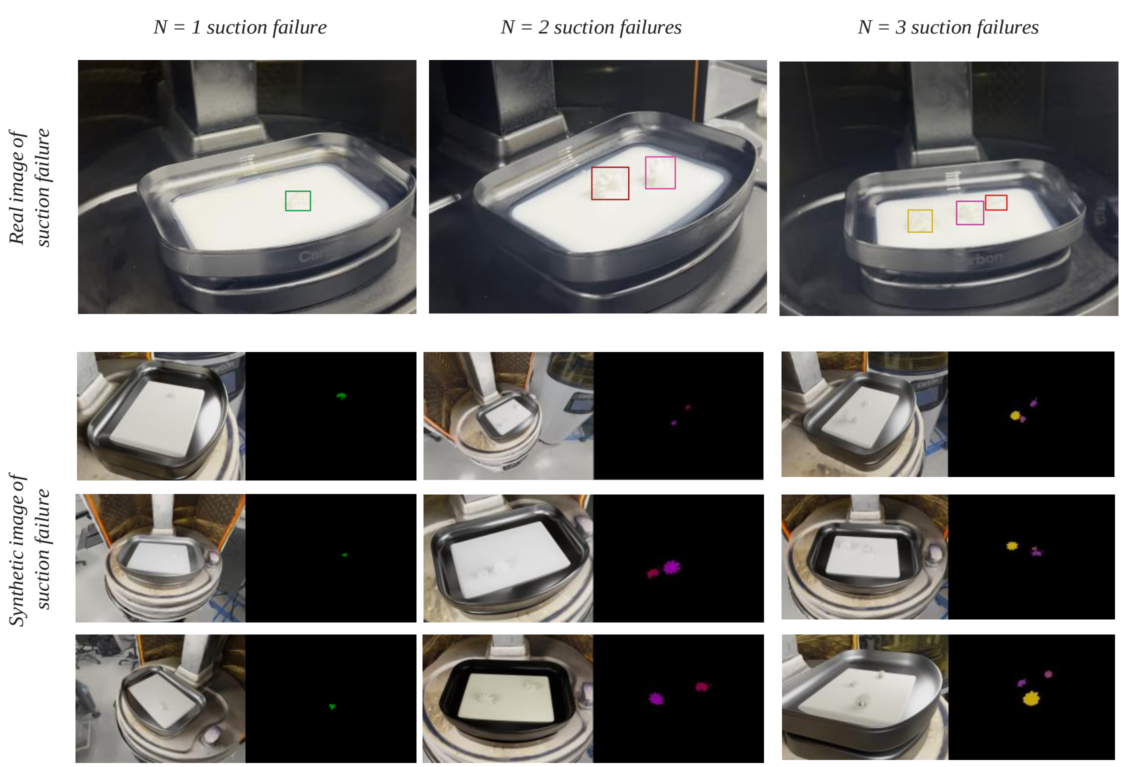
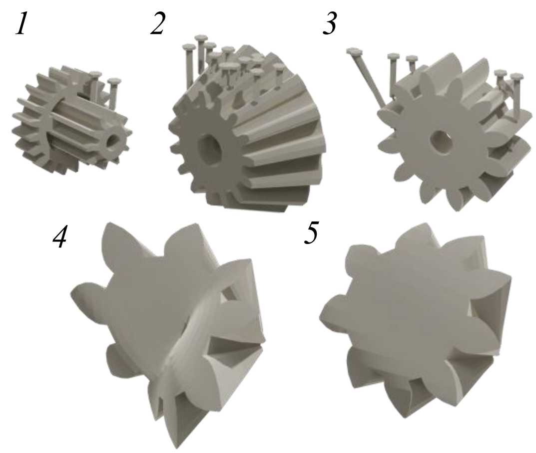


Fig. 4. Example real (top) and synthetic (bottom) images of print defects, with the latter generated from the user CAD model and a virtual scene of the 3D printer in Blender, and used to fine-tune a custom object detection model from a pre-trained YOLOv5 network. Segmentation masks for one (left), two (middle), or three (right) failures in a single image are shown to the right of their corresponding synthetic images.

2.2 Object detection model system evaluation

For evaluation purposes, we assess the performance of our object detection model in not only detecting suction-related failure, but also in discerning which of potentially multiple prints, and designs, have failed. To that end, we select several designs from the Thingi10k online repository [42] – representing real-world 3D printed designs – to train our model. To test our model's ability to discriminate between similar designs, we select variants of a similar design class, mechanical gear components, shown below:



While we train our object detection model on synthetic data for the superior scalability reasons outlined above, we evaluate the performance of our object tracking model on real-world data. We obtain a sample of suction-related print failures, specifically by artificially perturbing machine parameters and reducing support volumes, increasing the diversity of these real-world data by randomizing their position in the vat. Qualitative results of our defect detection model are shown in **Figure 5**, and quantitative training metrics in **Table 1**. As the number of design classes increases, the performance of our model drops, not unexpectedly, as shown in the confusion matrix corresponding to the five test designs shown in **Figure 6**. This suggests that a larger number of design classes pose greater challenges for accurate object tracking.

Fig. 5. Qualitative indications of the performance of our object detection model fine-tuned purely on $N = 500$ synthetically-generated images, for the task of detecting one defect (top), two defects (middle), or three defects (bottom). Numbers adjacent to detected design classes represent approximated confidence scores.

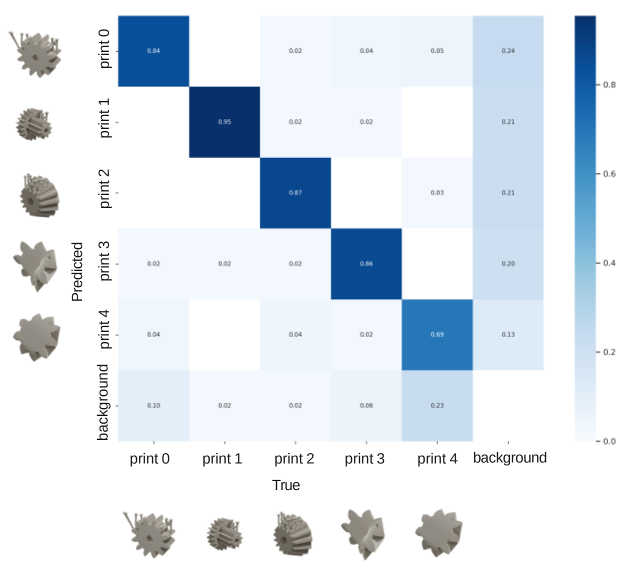


Fig. 6. Confusion matrix quantifying classification accuracy of the Yolov5 pre-trained object detection model fine tuned with 500 synthetically-generated print defect images with $N = 5$ design classes, specifically five five 3D models drawn from a subset of Thingi10k online repository (filtered by category "mechanical").

Additionally, recent research in the machine learning community suggests that the combination of large synthetic datasets with a smaller sample of real data can lead to the best performance [23].

To that end, we quantify the impact of injecting a small sample of real images of defective prints – specifically, 5 percent the size of our synthetic data – on model performance. Indeed, as shown in **Table 1**, this leads to significant improvements in model performance. We emphasize, however, that such real-world data is tedious to obtain, and potentially unfeasible for entirely new classes of designs, justifying our complementary simulation-guided data generation approach.

	mAP-50 (1 class)	mAP-50 (3 classes)	mAP-50 (5 classes)
Synthetic Data only (500)	0.821	0.668	0.489
Synthetic Data (500) + Real (25)	0.971	0.918	0.730

Table 1. Effect of design class count on tracking accuracy. mAP-50 scores are shown for object detection models evaluated on real data for varying numbers of classes.

	mAP-50
All DR	0.903
No material textures	0.885
No lighting conditions	0.811
No object pose	0.816
No camera pose	0.112

Table 2. Ablation study showing the effect of removing various domain randomization (DR) parameters on object detection model performance. Models trained with $N = 1000$ synthetic images.

Finally, we evaluate the importance of various simulation domain randomization factors in our synthetic data generation pipeline on detection accuracy. To do so, we perform ablation studies where we systematically eliminate one domain randomization parameter and assess the impact on model performance. All experiments performed with 1000 synthetic training sample images and pre-trained models fine-tuned for 50 epochs. The results are shown in **Table 2**. Here it is clear that camera pose is of critical importance, as expected, with model pose and lighting conditions also important to performance. This is in line with recent findings from the machine learning community [31].

Our detection model is trained to be orientation-agnostic, detecting objects from a variety of distances and orientations. However, our model does depend on having access to a clear line of sight to the printer vat, which may not be possible if the printer build platform obstructs view of potentially delaminated objects. To circumvent this, we can modify our print scripts to incrementally lift and lower the build platform during printing to provide access to unobstructed views; the frequency of such interruptions to printing is a balance between speed and resolution of defect detection.

3 Simulation visualization: augmented reality interface

In addition to alerting the user that a print failure has occurred, we also utilize our simulation framework to provide guidance on correcting such errors in future prints. To achieve this, we leverage recent advances in industrial augmented reality, which has seen growing popularity as a tool to overlay digital information on physical assets for real-time monitoring and user interaction [28, 29], including with 3D printers of numerous kinds [26, 1, 6] along with other digital fabrication systems [36, 13]. In general, such systems have enabled greater levels of interactivity and intuitiveness in the design and fabrication process compared with existing CAD/CAM workflows [40, 37, 10].

To enable similar interaction with simulation results for our system, we leverage the Unity Game Engine (version 2021.3.25f), a widely used cross-platform game development tool that provides a comprehensive suite of features for creating interactive and immersive experiences [34]. We briefly describe our system implementation for our interactive simulation tool is as follows. We load our simulation results with variable print parameters into the Unity AR scene environment, such that the user can readily toggle through them in real-world visualizations prior to printing, including by interactively zooming in and panning around the to-be-printed model in its accurate physical context.

To realistically overlay our numerical simulation data in the physical printer environment, we track the pose (position and orientation) of both the printer platform and vat; specifically, the user aligns a virtual overlay of the outline of the printer component with the real-world object (**Figure 7**). Tracking of the build platform and vat of the printer is theoretically straightforward, as exact CAD models are available. However, as these black anodized aluminium printer components themselves do not offer many readily trackable features, we improve tracking by endowing both printer components of with fiducial markers, specifically Aruco markers. Once tracked, we use their positional information to realistically overlay simulation results, organized as children of their respective printer components in the AR scene. This ensures that as the user pans around the printer, the pose of the virtual simulation data remains consistent with the real-world printer. We develop our tracking module with Vuforia (version 10.15.4) developed by PTC Inc., an augmented reality (AR) software development kit (SDK) [27].

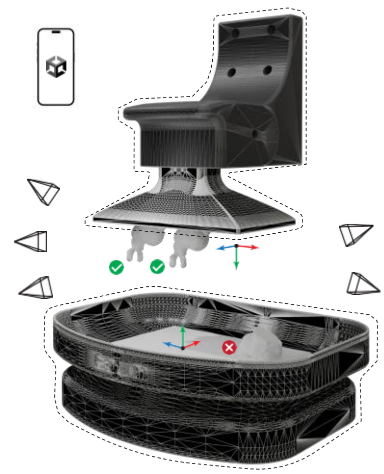


Fig. 7. Summary of procedure for tracking and pose estimation for AR simulation overlay, implemented in Unity3D, with tracked printer elements (platform and vat) highlighted.

3.1 AR simulation virtual overlay

Two general cases of example user interactions with our simulation data are shown in **Figure 8** and **Figure 9**. Each visualization illustrates a snapshot of a different simulation view in the AR interface, showing effective tracking whereby the simulation pose remains consistent despite changing user camera orientations in real world. The first case, illustrated in **Figure 8**, represents the case of when the user encounters a delamination defect, such as one detected in our module outlined in the previous section. Here, the user can align simulations of suction forces during printing with the failure manifestation, helping to explain where defects occurred and at which slices print parameters should be adjusted in future jobs. The second case, illustrated in **Figure 9**, shows the case of when the user needs to align a new print for injection 3D printing [18], and select injection rates to administer to offset suction. Here, rather than interfacing with an external CAD program, as is typical, the user can view simulation results directly on the platform where printing will occur, potentially aiding in dimensioning and object positioning and rotation prior to printing.

3.2 AR simulation application scenarios

We outline several potential applications of our interactive simulation for enhancing resin 3D printing workflows:

1. *Pre-print planning*: In the pre-print planning stage, it can be non-intuitive, especially for novice users untrained with CAD systems, to understand how the design and orientation of a 3D printed object impacts its printability. By overlaying fluid dynamics simulations

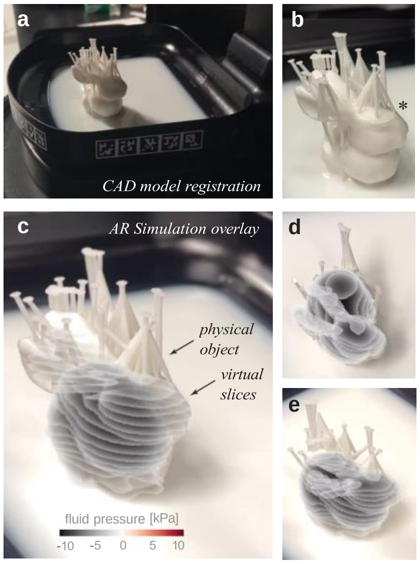


Fig. 8. Example simulation overlays for a print defect, indicated by asterisk, starting with CAD model registration of the printer vat (a-b), and alignment of the virtual slice simulations with the physical model from varying user viewpoints (c-e).

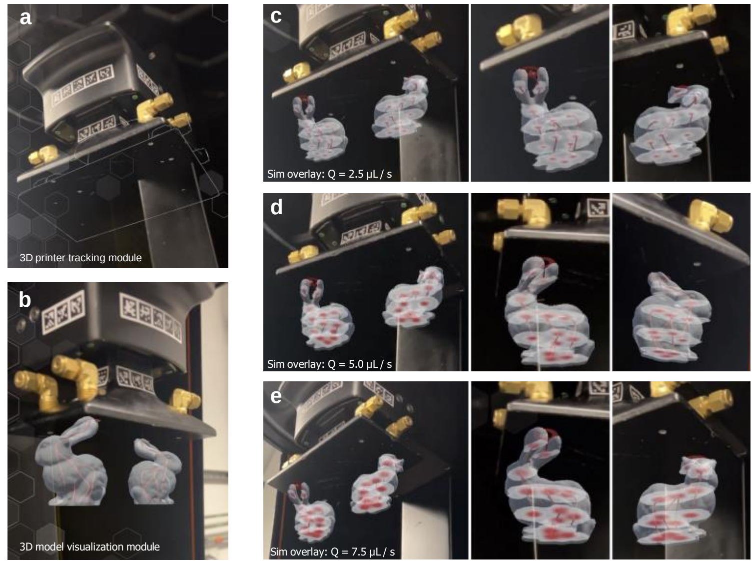


Fig. 9. Example simulation visualizations for injection printing, starting with CAD model registration of the printer platform (a), which is used to accurately depict fluidic networks for iCLIP printing aligned to build platform ports (b). The user can also visualize simulation results of fluid pressures at varying injection rates of $2.5\mu Ls$, $5.0\mu Ls$ (d), and $7.5\mu Ls$ (e), shown for a variety of camera viewpoints.

on a rendering of the to-be-printed object prior to production, obvious design flaws can be highlighted to the user and caught before printing.

2. *Real-time failure alerts*: During the printing process itself, if suction-related print failures do occur, typically users of vat 3D printers only are made aware of the result when they return to the printer. With our object detection tracking system, and with a simple RGB camera without need for calibration, users could be alerted to failures, saving time and material.

3. *Post-print debugging*: Even if suction-related failures are caught, it can be challenging for users, especially novice ones, to understand how and why the printing process failed. By directly overlaying simulation results on the print failure, as shown in **Figure 8 - 9**, we provide data that can facilitate the debugging process. In doing so, we allow designers to modify their designs, or print orientations and support structures, more intelligibly with the use of simulation data feedback overlaid on the real physical print result.

4 Limitations and Future Directions

Here, we report our ongoing development of an interactive simulation platform for predicting and visualizing suction-related defects in resin 3D printing. We furthermore demonstrate several ways by which users can interact with our simulation predictions in real-time and in the context of a real-world 3D printer. While we demonstrate our tool's object tracking capabilities – which rely upon the automation of simulation-generated synthetic data – can generalize to real-world data, we note several important limitations. First, our object tracking model breaks down when the resemblance of defective models to their corresponding CAD model declines. Moreover, if designs bear a high degree of similarity to one another, we observe our object detection model to incorrectly track, and confuse, their respective locations. Finally, while we show our system is able to effectively detect delamination failures, failures may also occur whereby the object remains attached to the platform, albeit deformed. In future, we will seek to more closely integrate our machine learning tracking capabilities with our interactive simulation tool for more seamless user experiences and for on-device performance. Currently our system employs iOS-based mobile devices; novel hardware interfaces for augmented and mixed reality applications, including head-mounted displays (HMDs), promise to further enhance the interactivity of our approach.

References

[1] Hirotaka Asayama, Daisuke Iwai, and Kosuke Sato. "Diminishable visual markers on fabricated projection object for dynamic spatial augmented reality". In: *SIGGRAPH Asia 2015 Emerging Technologies*. 2015, pp. 1–2.

[2] Blender Foundation. *Blender*. https://www.blender.org/. Version 3.3.0, Accessed: June 20, 2023. 2023.

[3] Steve Borkman et al. "Unity perception: Generate synthetic data for computer vision". In: *arXiv preprint arXiv:2107.04259* (2021).

[4] Douglas AJ Brion and Sebastian W Pattinson. "Generalisable 3D printing error detection and correction via multi-head neural networks". In: *Nature Communications* 13.1 (2022), p. 4654.

[5] Douglas AJ Brion and Sebastian W Pattinson. "Quantitative and Real-Time Control of 3D Printing Material Flow Through Deep Learning". In: *Advanced Intelligent Systems* 4.11 (2022), p. 2200153.

[6] Yi Cai, Yi Wang, and Morice Burnett. "Using augmented reality to build digital twin for reconfigurable additive manufacturing system". In: *Journal of Manufacturing Systems* 56 (2020), pp. 598–604.

[7] Flavia Dalia Frumosu et al. "Online monitoring for error detection in vat photopolymerization". In: *International Journal of Computer Integrated Manufacturing* (2023), pp. 1–18.

[8] J Giannatsis, V Dedoussis, and D Karalekas. "Architectural scale modelling using stereolithography". In: *Rapid Prototyping Journal* 8.3 (2002), pp. 200–207.

[9] Guo Dong Goh, Nur Muizzu Bin Hamzah, and Wai Yee Yeong. "Anomaly detection in fused filament fabrication using machine learning". In: *3D Printing and Additive Manufacturing* (2022).

[10] Ammar Hattab and Gabriel Taubin. "Rough carving of 3D models with spatial augmented reality". In: *Proceedings of the 3rd Annual ACM Symposium on Computational Fabrication*. 2019, pp. 1–10.

[11] Oliver Holzmond and Xiaodong Li. "In situ real time defect detection of 3D printed parts". In: *Additive Manufacturing* 17 (2017), pp. 135–142.

[12] You-Min Huang and Cho-Pei Jiang. "On-line force monitoring of platform ascending rapid prototyping system". In: *Journal of materials processing technology* 159.2 (2005), pp. 257–264.

[13] Yunwoo Jeong, Han-Jong Kim, and Tek-Jin Nam. "Mechanism perfboard: An augmented reality environment for linkage mechanism design and fabrication". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, pp. 1–11.

[14] Mohammad Farhan Khan et al. "Real-time defect detection in 3D printing using machine learning". In: *Materials Today: Proceedings* 42 (2021), pp. 521–528.

[15] Kavin Kowsari et al. "Photopolymer formulation to minimize feature size, surface roughness, and stair-stepping in digital light processing-based three-dimensional printing". In: *Additive Manufacturing* 24 (2018), pp. 627–638.

[16] Cézar B Lemos et al. "Convolutional neural network based object detection for additive manufacturing". In: *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE. 2019, pp. 420–425.

[17] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.

[18] Gabriel Lipkowitz, Eric Shaqfeh, and Joseph Desimone. "Paraflow: A Computational Design Tool for Support-free Multimaterial 3D Printing". In: *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. 2023, pp. 1–8.

[19] Gabriel Lipkowitz et al. "Digital Microfluidic Design for Injection Continuous Liquid Interface Production of 3D Objects". In: *2022 International Solid Freeform Fabrication Symposium*. 2022.

[20] Li Liu et al. "Deep learning for generic object detection: A survey". In: *International journal of computer vision* 128 (2020), pp. 261–318.

[21] Samarth Mishra et al. “Task2sim: Towards effective pre-training and transfer from synthetic data”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 9194–9204.

[22] Matthias Müller et al. “Sim4cv: A photo-realistic simulator for computer vision applications”. In: *International Journal of Computer Vision* 126 (2018), pp. 902–919.

[23] Farzan Erlik Nowruzi et al. “How much real data do we actually need: Analyzing object detection performance using synthetic and real data”. In: *arXiv preprint arXiv:1907.07061* (2019).

[24] Yayue Pan et al. “Study of separation force in constrained surface projection stereolithography”. In: *Rapid Prototyping Journal* 23.2 (2017), pp. 353–361.

[25] Konstantinos Paraskevoudis, Panagiotis Karayannis, and Elias P Koumoulos. “Real-time 3D printing remote defect detection (stringing) with computer vision and artificial intelligence”. In: *Processes* 8.11 (2020), p. 1464.

[26] Huaishu Peng et al. “RoMA: Interactive fabrication with augmented reality and a robotic 3D printer”. In: *Proceedings of the 2018 CHI conference on human factors in computing systems*. 2018, pp. 1–12.

[27] PTC Inc. *Vuforia: Augmented Reality Software Development Kit*. https://www.ptc.com/en/products/augmented-reality/vuforia. Version version 10.15.4, Accessed: June 20,2023. 2023.

[28] Moritz Quandt et al. “General requirements for industrial augmented reality applications”. In: *Procedia Cirp* 72 (2018), pp. 1130–1135.

[29] Daniel Röltgen and Roman Dumitrescu. “Classification of industrial Augmented Reality use cases”. In: *Procedia CIRP* 91 (2020), pp. 93–100.

[30] Mohammad Javad Shafiee et al. “Fast YOLO: A fast you only look once system for real-time embedded object detection in video”. In: *arXiv preprint arXiv:1709.05943* (2017).

[31] Jonathan Tremblay et al. “Training deep networks with synthetic data: Bridging the reality gap by domain randomization”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 969–977.

[32] Shashank Tripathi et al. “Learning to generate synthetic data via compositing”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 461–470.

[33] John R Tumbleston et al. “Continuous liquid interface production of 3D objects”. In: *Science* 347.6228 (2015), pp. 1349–1352.

[34] Unity Technologies. *Unity: Game Development Platform*. https://unity.com/. Version 2021.3.25f, Accessed: June 20, 2023.

[35] Alyson Vanderploeg, Seung-Eun Lee, and Michael Mamp. “The application of 3D printing technology in the fashion industry”. In: *International Journal of Fashion Design, Technology and Education* 10.2 (2017), pp. 170–179.

[36] Christian Weichel et al. “MixFab: a mixed-reality environment for personal fabrication”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2014, pp. 3855–3864.

[37] Christian Weichel et al. “ReForm: integrating physical and digital design through bidirectional fabrication”. In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 2015, pp. 93–102.

[38] Terry Wohlers. *Wohlers Report 2022: 3D Printing and Additive Manufacturing State of the Industry*. Fort Collins, CO, USA: Wohlers Associates, Inc., 2022.

[39] Mingtao Wu et al. “Detecting malicious defects in 3D printing process using machine learning and image classification”. In: *ASME International Mechanical Engineering Congress and Exposition*. Vol. 50688. American Society of Mechanical Engineers. 2016, V014T07A004.

[40] Amanda K Yung, Zhiyuan Li, and Daniel Ashbrook. “Printy3D: In-situ tangible three-dimensional design for augmented fabrication”. In: *Proceedings of the 17th ACM Conference on Interaction Design and Children*. 2018, pp. 181–194.

[41] Diane de Ze
licourt et al. “Single-step stereolithography of complex anatomical models for optical flow measurements”. In: *J. Biomech. Eng.* 127.1 (2005), pp. 204–207.

[42] Qingnan Zhou and Alec Jacobson. “Thingi10k: A dataset of 10,000 3d-printing models”. In: *arXiv preprint arXiv:1605.04797* (2016).