# Opportunities for Utilizing Consumer Grade 3D Capture Tools for Insurance Documentation

Kevin Ponto[1*] and Ross Tredinnick[1]

[1*]Wisconsin Institute for Discovery, University of Wisconsin-Madison, 330 N Orchard St., Madison, 53715, WI, United States.

*Corresponding author(s). E-mail(s): kbponto@wisc.edu; Contributing authors: rdtredinnick@wisc.edu;

**Abstract**

Recent advances in consumer technology have enabled a new means for 3D capture of real world environments and objects through the adoption of depth sensors within tablets and mobile phones. While traditional methods to capture 3D objects are often cost-prohibitive, the ability to have commodity grade scanning technologies readily available creates new opportunities for 3D documentation. This paper showcases the opportunities for the utilization of 3D capture and AR technologies for insurance documentation. The benefits and challenges of using 3D capture given these emerging technologies are discussed. Finally, the paper introduces prototype applications for the full 3D modeling of a vehicle, detection of vehicle damage both from vehicular collisions and from hail.

**Keywords:** 3D Capture, 3D Scanning, Insurance Documentation, Consumer Technologies

## 1 Introduction

Traditional mechanisms of three-dimensional (3D) capture occur through operating survey grade 3D scanning equipment, such as terrestrial Light Detection and Ranging (LiDAR) scanners [1]. This technology generates data in the form of high resolution depth images, often paired with high resolution color images, that when combined, generate a 3D point cloud. A point cloud is a data type in which the positions and attributes (often colors) are stored without a given order. While this form of 3D scanning has been commercially available for 30 years, cost has limited mass adoption [2]. This limitation has existed despite the fact that up front costs have the potential to be offset by benefits in certain applications, such as crime scene investigation [3]. More recently, advances in consumer technology have enabled a new means for 3D capture of real world environments and objects through the adoption of depth sensors within tablets and mobile phones. These sensors can accurately measure distances from a device. Coupling these sensors with methods for 3D tracking of position and orientation of these devices, enables anyone to 3D capture objects and spaces, creating a new generation of 3D scanning applications (as shown in Figure 1).

**Fig. 1** An example image showing a 3D capture of an automobile, including sampling locations (orange outlined photographs)

Augmented Reality (AR) is a second emerging technology that has recently moved into a consumer space. AR refers to the merging of a live view of the physical, real world with context-sensitive, computer-generated images to create a mixed reality [4]. AR has seen a surge in consumer use in recent years due to its support on mobile devices and through popular applications such as Pokemon Go [5]. As this technology has continued to evolve, new opportunities for these types of technologies have been seen in many application areas [6].

This manuscript will highlight a number of novel prototype applications that the research team has developed that combines 3D capture and AR technology with the field of automobile insurance documentation. In the remainder of this manuscript, these opportunities for the utilization of 3D capture and AR technologies for insurance documentation are showcased. Techniques for generalized 3D capture of automobiles using consumer available depth sensors are introduced. Finally, opportunities for specific automobile insurance related AR and 3D capture applications are explored.

# 2 Prior Work

Existing software applications for automobile insurance currently range from documentation and estimation of accident damage to detecting fraudulent claims [7, 8]. Several existing techniques capture 2D images using traditional cameras as a part of the data collection process for input into software applications [9–11]. In the case of using 2D images for documentation, due to the narrow field of view in resultant photographs, along with the potential need to take close-up photographs of damaged areas, the greater context and surrounding environment of the captured location can easily be lost [12]. These images are therefore prone to discrepancies during insurance review and assessment [13].

Prior AR work in the automotive industry has mostly centered on applications related to either "out-of-car" experiences, such as aiding in the automotive manufacturing process (maintenance, design, inspection, training, etc.), or "in-car" experiences, such as providing heads-up displays within the field of view of a driver) [14]. The main goals of "in-car" systems are to both enrich the driving experience, as well as prevent accidents by increasing the driver's attention and perception. This could come in the form of warnings that show up on the windshield, when a driver is getting too close to an obstacle. Care must be taken in the design of these "in-car" AR applications to not further distract a driver, and many are actively researching this topic [15–17]. One example of an "out-of-car" application matches whether an automobile part correctly matches with the intended construction design of that part, to help guarantee correspondence [18]. Prior to recent advances in mobile 3D tracking technology, researchers developed a markerless tracking system for AR apps in the automotive industry [19]. For more details on AR in the automotive industry, existing researchers have previously summarized additional work in a thorough review [20]. To note with these AR applications, they are not developed directly with a means to benefit insurance companies, but may so indirectly. The presented work will focus on AR applications that could be developed to directly benefit automobile insurance companies.

Prior work surrounding 3D scanning in the automotive industry has been less frequent, and mostly centered on applications for modeling vehicle crashes [21–24]. The benefit behind being able to 3D scan and model vehicle crashes comes up in the situation where a later discrepancy may exist surrounding the crash. This concept helps reduce risk for the automobile insurance companies as it reduces potential uncertainty surrounding the incident. With a a high resolution 3D model of the crash, those involved can reference the 3D model to potentially resolve questions surrounding the incident. Unfortunately, until recently, the cost of 3D scanning technology for performing scans of car accidents has limited mass adoption; however, as will be presented below, mobile 3D scanning technology is quickly changing this limitation [25].

Additional previous work has sought to automatically detect dents on car bodies [26–30]. These approaches all sought to combine 2D images with Artificial Intelligence (AI) algorithms for automatically detecting dents in vehicle bodies, but did not incorporate 3D information that now exists from mobile sensors. Existing commercial software solution adoptions have aimed to utilize AI in the claims process, but again most of these solutions focus on 2D images as opposed to 3D information.

# 3 Technological Methods and Challenges

This section examines the opportunities and challenges for utilizing consumer grade 3D capture devices for automobile insurance companies.

## 3.1 Terrestrial LiDAR

Terrestrial LiDAR scanners are used as a highly accurate way to capture a space. Commonly, LiDAR scanners capture depth and color information in a panoramic fashion. In means of making a 3D model, the scanner is moved around the environment to fill in information that is occluded from other scan locations. These scans are then registered together in a unified space to create a high fidelity model. While these scanners are capable of producing high quality results, their cost is often a limiting factor towards their adoption [3]. With the new ability to capture 3D automobiles with mobile devices, it is of interest to compare results of this with traditional 3D scanning hardware, such as a terrestrial LiDAR scanner. So far researchers have found that mobile LiDAR scanning is accurate to within 1-3 cm of industry-grade terrestrial LiDAR scanners [24]. This work surmised that this accuracy is close enough to reliably 3D capture automobile accidents.

## 3.2 Augmented Reality

One requirement for any AR experience is to create and track a correspondence between the physical space the user inhabits and a virtual space where you can model visual content. In means of compositing virtual content on a real environment, a device must understand its position and orientation in space (i.e. pose estimation) and the surrounding environment [31, 32]. Different algorithms exist for calculating the position of a device within 3D space using information from the environment, with simultaneous localization and mapping (SLAM) serving as a common approach [33]. Mobile AR software, such as ARKit and ARCore make use of a variant of SLAM, known as Visual-Inertial Simultaenous Location and Mapping (VI-SLAM) [34] [35]. These methods utilize tracking of natural features to determine structure of the scene and the pose of the device. One step in the process of SLAM is to determine an estimation of depth of the current environment from the perspective of a camera viewpoint.

Prior methods for accomplishing the depth acquisition step on mobile devices typically used 2D camera images processed over consecutive frames and / or images obtained from disparity between stereo cameras [36–39]. These depth images can be utilized to create occlusions between physical and virtual objects. In 2020, Apple released LiDAR sensing hardware on their iPad and iPhone. This sensor provides a level of depth sensing that is quantifiable in terms of accuracy and that can be obtained in a single frame. Prior research has shown that the inclusion of LiDAR sensors for depth acquisition has improved accuracy and allows for a more robust SLAM method, thus allowing for more accurate and reliable pose estimation (also known as six degree of freedom tracking) of the device in space [40]. This improved tracking can create immersive AR experiences: a virtual object can appear to stay in the same place relative to the real world, even as the user tilts the device to look above or below the object, or moves the device around to see the object's sides and back.

Despite the improved tracking ability that LiDAR affords, mobile-based AR technologies are still prone to a problem commonly referred to as drift, which causes accuracy of the tracking to degrade over operation time of the device. Drift can occur for many reasons such as noise, poor lighting or changes to lighting conditions, gravity acceleration, and moving across large distances while operating the device [41–43]. A common result from drift is that virtual objects will become misaligned with prior real world locations where objects had been positioned. For example, if a user places a virtual object within the AR display's field of view such that it looks positioned on a table that exists in the real world, and then the user walks some distance, and orients the device away from the table, and then returns and tries to view the object, if the object has changed location with the AR display's field of view, this would be an example of drift.

## 3.3 3D Capture

While Apple promoted the addition of the LiDAR sensor to their mobile hardware as a feature to improve AR applications, a second ability that the sensor affords is the ability for improved 3D capture of real world spaces and objects. This ability becomes possible thanks to the LiDAR sensor as well as the aforementioned 3D tracking capabilities of the devices. The addition of the LiDAR sensor first occurred in the iPad Pro 2020 and the iPhone 12 X Pro.

Together with the LiDAR sensing hardware, Apple added software support for accessing LiDAR data through ARKit, Apple's AR software infrastructure. ARKit provides an interface for obtaining LiDAR data as a depth image. The depth image is formatted where each pixel contains a single floating point value that is the distance from the sensor of any real world objects within the field of view of the LiDAR sensor at time of capture. The LiDAR sensor within iOS devices has a 256 wide x 192 high resolution for a total of up to 49,152 depth samples per capture. Samples are limited to a distance of 8-10 meters away from the sensor. In addition to this depth image, ARKit also provides a “confidence image”, which is an image of the same dimensions (256x192) where each pixel is a value 0, 1, or 2, with the value 2 serving as most confident, and 0 as least confident in terms of distance accuracy of the corresponding depth pixel. In addition to these two images, the traditional color camera on these devices can be used to capture color information, to pair with the depth data. Given a series of device poses with their associated depth and color images, there are various methods that can be used to generate a 3D model. A few approaches are described below.

One simple way to create a 3D model from this setup is to unproject captured depth image data back out into 3D space [44]. This is done via undistorting the camera image, aligning the color image with the depth to make a view aligned set of points and unprojecting these points into space. The resultant dataset, commonly known as a point cloud, consists of an unordered collection of points that stores 3D positions and colors. This approach is generally used for high fidelity 3D scanning equipment, such as terrestrial LiDAR, in which errors in depth at the time of capture are minimal [45].

Point clouds can be rendered using computer graphics application programming interfaces, such as OpenGL and DirectX, or with 3D content creation software such as Unity. Others approach point clouds by seeking to convert them into polygonal 3D meshes, and for certain applications or use-cases this may be a proper route to pursue. A downfall of converting to a polygonal mesh is that in the case of real world environments, detail of particular objects such as vegetation, clothing, and other objects of complex shapes may become compromised [46].

One issue when using this approach with consumer grade hardware is misalignments in tracking data can quickly create visual artifacts as shown in Figure 2 Left below. These types of misalignments will show up as groups of points that appear to be floating in space. These misalignments occur due to the aforementioned drift problems as well as potential complex lighting reflections that can exist on vehicles. One solution to this is to realign captures in a post processing step such that one can compare each scan versus all other scans. When misalignments are detected they can then be realigned via processes, such as iterative closest point [47]. This solution can solve small scale tracking issues as shown in Figure 2 right below. Unfortunately, this corrective process may require fine tuning of algorithm parameters to achieve good results, and overall is unideal for a mobile 3D capture application.

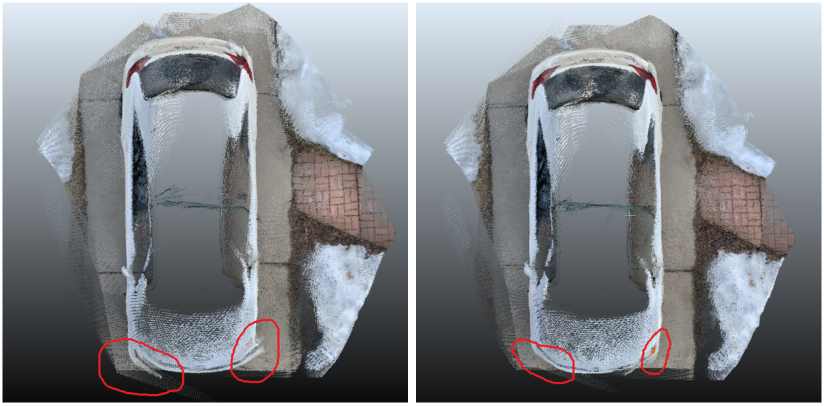

**Fig. 2** Misalignments of single captures in relation to the rest of the model. Right: Corrected scans after Generalized Iterative Closest Point algorithm)

The Truncated Signed Distance Function (TSDF) technique for 3D reconstruction also works from a set of depth images acquired over a series of frames, but performs a slightly different approach from the point unprojection method described above [48]. The TSDF algorithm traditionally works by breaking up 3D space onto a 3D grid at a user-specified resolution. All grid cells are initialized with a non-zero value. On each frame (or at a user-specified frequency), the current frame's acquired depth image's samples are unprojected into 3D space and then the algorithm determines for each point in the resultant point cloud which grid cell the point lands in. The 3D locations of these samples are compared against the 3D center points of the grid cells they land in and the differences in 3D distance are stored in each grid cell within the volume [49]. These differences are averaged each frame, and the result is a progression of grid values towards zero for real world objects and spaces that are captured.

From this information, the position of surfaces can be found. Using algorithms such as marching cubes, a 3D mesh of the model can be constructed [50]. The major advantage of the TSDF approach is that samples are averaged over time, thereby mitigating sensor noise and tracking errors. However, the challenge with this approach is that a dense 3D volume is required to store all of the samples. This means that memory requirements are dependent on the size of the space, and detail to be captured can quickly exceed the abilities of mobile devices. For the research team's prototype implementation of this technique, and to help alleviate these memory issues, the team utilized a series of small, sparse voxel grids to only compute surfaces where they were detected. This process is described in further detail below in section 4.

## 3.4 Common Challenges

A common challenge that exists for the two described mobile 3D capture techniques is known as loop closure, or the problem of guaranteeing that when moving to a new space while scanning and then returning to a previously scanned location, that newly captured data samples will lie in the same 3D location as the previously captured samples [51]. Loop closure will fail when the device thinks it's in a physically different location or orientation than it previously had been even though the user may have the device physically positioned and oriented at the exact same previous location. This problem is similar to that mentioned in Section 2 regarding placement of objects for AR and is an indirect result of drift.

# 4 Applications

In discussions with representatives in the insurance field, the concept of 3D modeling vehicles has a multitude of applications. Some applications include modeling for accident documentation, detection of vehicle damage, and specifically, detection of hail dents. The following section will discuss our approaches to the development of prototype applications in these areas.

## 4.1 Accident Documentation 3D Modeling


**Fig. 3** User interface on iPad for 3D capture for accident documentation application

3D capture of accident scenes have the benefit of providing a three dimensional "snapshot" in time, in the form of a 3D model, of the incident. This provides the ability for persons involved to be able to go back and reference this model in the event that any questions surface after the fact (for example, in the court of law). While previous work in this area focused on analyzing 3D scanning of these scenes via terrestrial LiDAR scanners and a custom made PanoScan point gun portable 3D scanner, the addition of LiDAR to mobile iOS allows for 3D accident capture with these more ubiquitous devices [3].

The research team has developed a 3D scanning application for automobile accident capture compatible with the iPad Pro 2020, iPad Pro 2021, or iPhone 13 using ARKit. The user interface of the prototype 3D capture app is shown in Figure 3. Initialization of ARKit's 3D tracking occurs at startup of the application. A user can choose a desired capture resolution via the top slider, and also adjust the desired capture bounds via the vertical slider on the right side. The app depicts a greyscale view over areas that won't be captured as a result of manipulating the capture bounds slider. The red button begins capture, at which point a user physically walks around the object or throughout a space, aiming the device at the desired areas of capture. Individual depth and color captures occur at a rate of one per second. When the red button has been pressed and the app is actively 3D capturing a space, captured data is tinted green, while capturing is successfully occurring, or tinted red, if a capture is being skipped. The app has built in logic to skip capturing of certain frames based on certain conditions, such as sudden changes in lighting, poor lighting, or sudden unintentionally quick movements of the mobile device. When pressing the red button again, the app will process the captured data on-device and write out a 3D point cloud model in XYZ ASCII format. The XYZ format is a common, basic format for 3D point cloud models, consisting of a list of x,y,z positions, and r,g,b colors. In addition, the app can optionally show a preview of the capture results immediately after the writing of the data. The gray button can be used to perform a single capture, as opposed to it continuously capturing.

The team implemented a modified TSDF approach for capture of 3D data (Fig. 4). The team built upon the traditional TSDF approach and performed an optimization such that the application dynamically allocates "mini volumes" instead of the naive approach of implementing a single large volume. Conceptually there is still a dense volume grid, but not all memory is allocated at once, instead only mini volumes for which depth samples have hit in the scene are allocated. This allows the algorithm to ignore having to allocate memory for empty space and follows from previous approaches [52] allowing the capture of resolutions down to one millimeter of precision. This process would not be possible using traditional methods that store data in a dense format, as the data need would greatly exceed the current memory available on mobile GPUs. After a scanning session ends, the app then walks back through the volume to check for values that are near zero and writes this information out as a 3D point cloud. A result of the team's implementation of the TSDF technique is shown in Figure 5.

```
Obtain Image color, Image depth, Image conf,
Matrix4x4 camIntrinsInv, Matrix4x4 camToWorld for
current frame.

Set color, depth, conf, camIntrinsInv, camToWorld
as input to compute shaders.

1st Compute Shader Pass:
For each depth pixel with max confidence,
unproject to point P in 3D space using camIntrinsInv,
camToWorld.

Determine which 32x32x32 subgrid g within the larger, user
defined grid that P lands in

Readback subgrid results, create set

Allocate memory for subgrids that have not been yet,
32x32x32xsizeof(half) for TSDF value,
32x32x32xsizeof(uint) for color

2nd Compute Shader Pass:
For each depth pixel with max confidence,
unproject to point P in 3D space using camIntrinsInv,
camToWorld.

Lookup which 32x32x32 subgrid G, P lands in

In previously allocated memory, update and store TSDF
value and color.
```

**Fig. 4** Pseudocode for TSDF capture algorithm

As a part of the post-processing tools accompanying the app, a user can also show individually captured color images together with the captured 3D scan in the original context from a scanning session (Figure 6). The post-processing tools were developed for the Unity 3D content creation platform, which has become a de-facto standard application for authoring 3D content. The research team believes that this can be useful for obtaining a better understanding of the overall context of a 3D space as compared to just looking at the individual photographs.

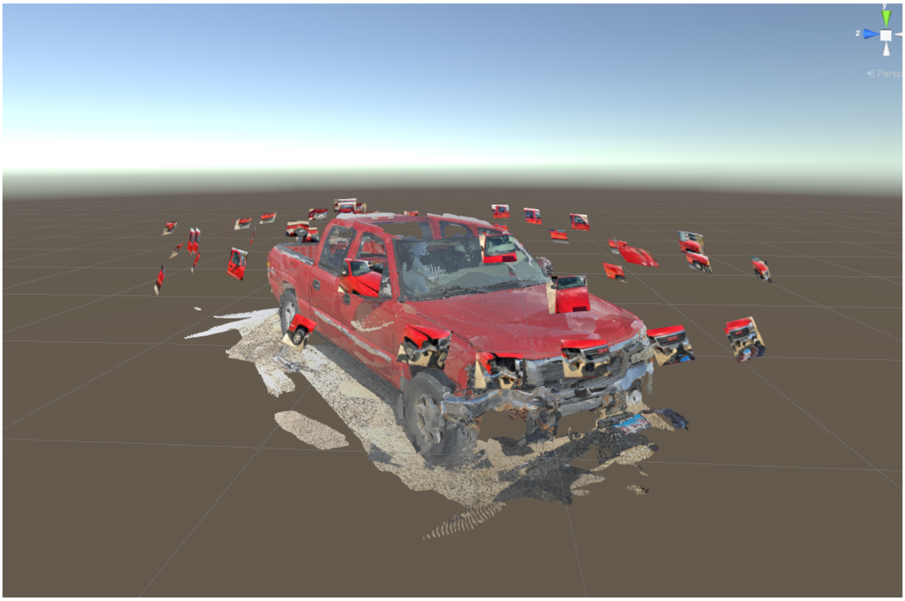

**Fig. 6** An example 3D capture of a front-end collision accident of a GMC truck using Direct Point Projection, loaded within Unity. Individual captures are shown as rectangles at the scale of the iPad. The color photo captured for each sample is shown in the original 3D position and orientation from time of capture

## 4.2 Damage Detection

Beyond 3D scanning to generate a 3D model of an automobile accident, another helpful application where 3D capture could benefit is in the assessment of damage on a single vehicle. 3D capture of a damaged vehicle has the potential to gather additional information beyond traditional 2D photographs. Furthermore, this information could be combined with machine learning and artificial intelligence algorithms to automatically assess damage costs. One specific question that claims adjusters seek to answer early on in their process is whether a vehicle is totaled. Determination of whether a vehicle is totaled is important due to the potential cost savings for insurance companies, as further time spent by personnel on the particular incident can be reduced. New tools to help more quickly and accurately assess and determine a “totaled” state are therefore attractive for automobile insurance companies.

Using the concept that most vehicles are symmetrical down their longest axis, an early prototype for determining vehicle damage was created as shown in Figure 7 Left. Given this assumption, the team developed two algorithms for detecting damage on a vehicle. The first approach divides each half of space down the center axis into a 3D grid, and compares the existence of point cloud information (generated from the app described in Section 5.1) between corresponding grid cells across the axis of symmetry. Areas without corresponding information across the axis highlight as green (Figure 7 Middle). The technique suffers from needing to determine an appropriate grid density in order to obtain visually accurate results. It also suffers from the fact that if the vehicle has a slight rotation or tilt to it (perhaps being on uneven ground), requires minor adjustments to the 3D grid so that it aligns appropriately for the calculation to work.

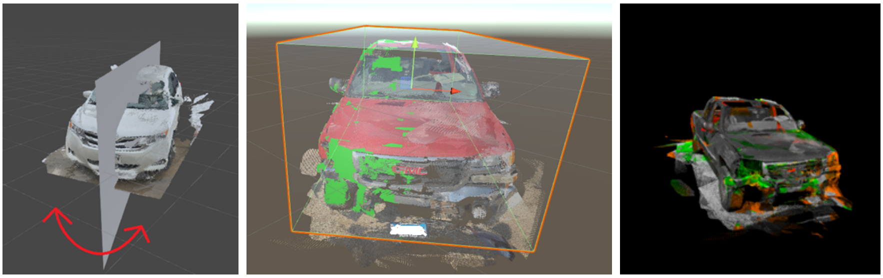

**Fig. 7** Left - theory behind damage detection prototype of a vehicle, due to the symmetrical nature of most automobiles, comparing the existence of 3D scanned data from one half of the vehicle to the other in 3D provides information on the existence of potential damage. Middle: Example visualization of 3D grid prototype. Right: Example visualization of GPU-shading approach for visualizing damage.

The second approach uses a computer graphics technique whereby instead of using a user defined grid and compute shader, the team took the approach of trying to compare two instances of the same scanned model, with one of them having been flipped along the axis down the center of the vehicle. Positional values were then compared using a GPU shading approach within Unity, whereby one vehicle’s world space positions are rendered from the current camera view into a texture, and upon rendering the flipped version of the vehicle, a lookup is done at the current pixel to compare the non-flipped position value, with the position value currently being rendered. The difference of these world space values are mapped to a color scale, green = less difference to red = most different. A resultant visualization showing this technique is shown in Figure 7 Right. This approach unfortunately suffers from needing to align one model to another, with this transformation typically being more than just a single rotation, rather involving a translation + rotation, due to the fact that the captured scan data is not guaranteed to be axis aligned.

Both damage detection approaches currently function as part of the research team's set of Unity post processing tools. To improve upon the described setbacks, the team anticipates incorporating a tracking marker, such as a centered QR code on the ground in front of the vehicle, as this would help realign the capture space to the natural axis of symmetry of the vehicle. This would allow for an easier comparison in the GPU-shading approach, and could also allow for a potential real-time (instead of post-process) damage visualization within an ARKit app as the vehicle is 3D captured from both sides.

## 4.3 Hail Detection

Due to the nature of vehicle damage from hail dents, their existence and damage extent can be difficult to discern with the naked eye. Furthermore, due to the reflective properties of vehicle surface paint in regards to color and lighting, hail dents become difficult to distinguish when analyzing only color images. For example, Figure 8 Left highlights difficult to see locations of hail dents on a vehicle. Due to these challenges, there is a strong desire within the automotive insurance agency to develop technology for improved detection of hail dents.

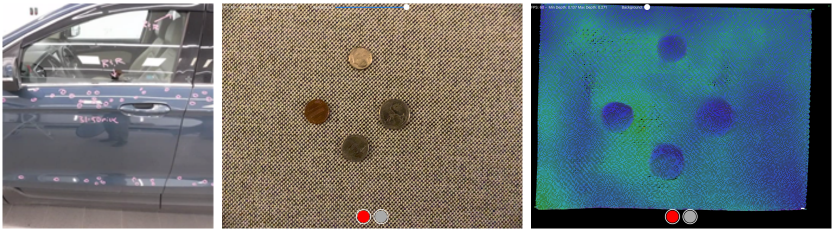

**Fig. 8** Left: Hail dents identified on a vehicle, circled in pink and very difficult to notice without the marks. Middle: Four coins laying on a flat surface. Right: Shows how the depth differences for coins on a flat surface are detectable using the integrated LiDAR scanning technologies in the iPad Pro 2020 model.

The goal of this work is to utilize the iPad's LiDAR scanner to highlight changes in depth in means of detecting hail damage. The prototype works by finding the maximum and minimum depth of the current view in real time. This is accomplished via analyzing each of the 49,152 LiDAR depth pixels. The app also calculates an estimated normal vector for each depth pixel of the current view. By mapping a combination of the normal vector and the depth to a color scale, visualizations can be created as shown in Figure 8 right. This initial prototype causes small objects such as the coins shown in Figure 8 middle to be distinguished without using any color information. Pseudocode for the hail detection algorith is provided in Figure 9. The mode is currently accessible via the UI on the app by toggling to a hail detection mode in the lower left corner (Figure 3). Upon changing to this mode, the user orients the iPad to bring the area of detection into view and then slowly moves the iPad towards the

```
Obtain  Image depth, Image conf, for
current frame

Compute Shader:
 Set depth, conf as input
 Set RWBuffer for max / min depth range as input
 Run compute shader to obtain depth range

Vertex Shader:
 Render fullscreen quad

Fragment Shader:
 Set RWBuffer for depth range (minDepth, maxDepth)  as input
 For each pixel (p), calculate screen space normal vector (n)
   if(conf(c) == maxConfidence)
   {
     d = depth(c)
     f = d - minDepth / (maxDepth - minDepth)
     //vary hue depending on depth
     hsvCol = float3(hue(saturate(f)), 0.9, 0.6)
     dc = hsvToRGB(hsvCol)
     fragColor = 0.2 * dc + 0.8 * n
   }
```

**Fig. 9** Pseudocode for Hail Detection algorithm

vehicle. While this approach works in principle, in practice it was found to be susceptible towards noise, leading towards inconsistent results for smaller hail damage. As the sensor technology continues to evolve, this approach is likely to provide more reliable results.

# 5 Discussion

While this paper presents prototypes of the utilization of augmented reality for 3D capture, there are other methods to create 3D models beyond those described in the document. One recent form of 3D reconstruction that has risen in popularity is known as Neural Radiance Fields [53]. Neural Radiance Fields use machine learning, in the form of multilayer perceptrons (MLP), to create a representation of a space using a set of photographs. Embedded in this representation is an inferred representation of the 3D model including the colors and reflectance. This form of modeling has gained a large amount of enthusiasm as complex environments can be reconstructed in ways previously not possible [54–57]. This newly derived model format as represented by MLPs has the potential benefits of a much smaller file size when compared to traditional model formats. However, the validity and verifiability of the underlying 3D data is an open research question. Despite this, these methods have shown a great potential for fast reconstruction of 3D objects. Beyond this, as the number of consumer devices that integrate LiDAR technologies has continued to grow, the technologies for 3D capture have become more readily available. As the barriers towards creating digital models of real-world objects and spaces are removed, the potentials for 3D data analysis both by humans and AI will only continue to expand [58].

# 6 Conclusion

This paper presents opportunities for utilizing consumer grade 3D capture tools for insurance documentation. While the quality of capture using current consumer grade LiDAR sensing technologies are much less than commercial LiDAR systems, the accessibility of these systems creates new possibilities for documentation and assessment. This document demonstrates prototypes of how these 3D capture technologies can be used for modeling and photo positioning, damage assessment, and hail detection. While these prototypes provide a proof of principle, for these methods to be used in a commercial application, more work would need to be done. As technology continues to evolve, better methods for tracking and depth and distance measures will likely become available on consumer devices. These improved devices will likely provide new opportunities for their integration into the insurance documentation process. As the barriers to create 3D models of spaces and objects are removed, one can imagine the transformative effect for both insurance clients and providers.

# References

[1] Fidera, A., Chapman, M.A., Hong, J.: Terrestrial lidar for industrial metrology applications: modelling, enhancement and reconstruction. In: XXth ISPRS Congress, vol. 5, pp. 880–886 (2004)

[2] Spring, A.P.: History of laser scanning, part 2: The later phase of industrial and heritage applications. Photogrammetric Engineering & Remote Sensing **86**(8), 479–501 (2020)

[3] Tredinnick, R., Smith, S., Ponto, K.: A cost-benefit analysis of 3d scanning technology for crime scene investigation. Forensic Science International: Reports **1**, 100025 (2019)

[4] Furht, B.: Handbook of Augmented Reality. Springer Science & Business Media, Berlin, Germany (2011)

[5] Rauschnabel, P.A., Rossmann, A., tom Dieck, M.C.: An adoption framework for mobile augmented reality games: The case of pokémon go. Computers in Human Behavior **76**, 276–286 (2017)

[6] Mekni, M., Lemieux, A.: Augmented reality: Applications, challenges and future trends. Applied computational science **20**, 205–214 (2014)

[7] Waqas, U., Akram, N., Kim, S., Lee, D., Jeon, J.: Vehicle damage classification and fraudulent image detection including moiré effect using deep learning. In: 2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1–5 (2020). IEEE

[8] Ghorbani, A., Farzai, S.: Fraud detection in automobile insurance using a data mining based approach. International Journal of Mechatronics, Elektrical and Computer Technology (IJMEC) **8**(27), 3764–3771 (2018)

[9] Li, Y., Dorai, C.: Applying image analysis to auto insurance triage: A novel application. In: 2007 IEEE 9th Workshop on Multimedia Signal Processing, pp. 280–283 (2007). IEEE

[10] Masingha, M.T.: Image processing based auto insurance claims fraud detections. PhD thesis (2021)

[11] Thonglek, K., Urailertprasert, N., Pattiyathanee, P., Chantrapornchai, C.: Vehicle part damage analysis platform for autoinsurance application. ECTI Transactions on Computer and Information Technology (ECTI-CIT) **15**(3), 313–323 (2021)

[12] Barrett, T.: Teaching about photography: Photographs and contexts. Art Education **39**(4), 33–36 (1986)

[13] Huetter, J.: Insurers testing ccc's a.i. smart estimate; product goes on sale early next year. Repairer Driven Communications (2018)

[14] Rohacz, A., Strassburger, S.: Augmented reality in intralogistics planning of the automotive industry: state of the art and practical recommendations for applications. In: 2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA), pp. 203–208 (2019). IEEE

[15] Fu, W.-T., Gasper, J., Kim, S.-W.: Effects of an in-car augmented reality system on improving safety of younger and older drivers. In: 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 59–66 (2013). IEEE

[16] Froehlich, P., Schatz, R., Leitner, P., Mantler, S., Baldauf, M.: Evaluating realistic visualizations for safety-related in-car information systems. In: CHI'10 Extended Abstracts on Human Factors in Computing Systems, pp. 3847–3852 (2010)

[17] Wiegand, G., Mai, C., Holländer, K., Hussmann, H.: Incarar: A design

space towards 3d augmented reality applications in vehicles. In: Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications, pp. 1–13 (2019)

[18] Nolle, S., Klinker, G.: Augmented reality as a comparison tool in automotive industry. In: 2006 IEEE/ACM International Symposium on Mixed and Augmented Reality, pp. 249–250 (2006). IEEE

[19] Lima, J.P., Roberto, R., Simões, F., Almeida, M., Figueiredo, L., Teixeira, J.M., Teichrieb, V.: Markerless tracking system for augmented reality in the automotive industry. Expert systems with applications **82**, 100–114 (2017)

[20] Boboc, R.G., Gîrbacia, F., Butilă, E.V.: The application of augmented reality in the automotive industry: A systematic literature review. Applied Sciences **10**(12), 4259 (2020)

[21] Pagounis, V., Tsakiri, M., Palaskas, S., Biza, B., Zaloumi, E.: 3d laser scanning for road safety and accident reconstruction. In: Proceedings of the XXIIIth International FIG Congress, vol. 8, p. 13 (2006)

[22] Tandy, D.F., Coleman, C., Colborn, J., Hoover, T., Bae, J.: Benefits and methodology for dimensioning a vehicle using a 3d scanner for accident reconstruction purposes. Technical report, SAE Technical Paper (2012)

[23] Lyu, N., Huang, G., Wu, C., Duan, Z., Li, P.: Modeling vehicle collision angle in traffic crashes based on three-dimensional laser scanning data. Sensors **17**(3), 482 (2017)

[24] Desai, J., Liu, J., Hainje, R., Oleksy, R., Habib, A., Bullock, D.: Assessing vehicle profiling accuracy of handheld lidar compared to terrestrial laser scanning for crash scene reconstruction. Sensors **21**(23), 8076 (2021)

[25] Ramasubramaniam, K.S., Bhat, R.: Lcar—low cost augmented reality for the automotive industry. In: 2018 IEEE International Conference on Consumer Electronics (ICCE), pp. 1–3 (2018). IEEE

[26] Lilienblum, T., Albrecht, P., Calow, R., Michaelis, B.: Dent detection in car bodies. In: Proceedings 15th International Conference on Pattern Recognition. ICPR-2000, vol. 4, pp. 775–778 (2000). IEEE

[27] Chen, H.: Automatic dent detection on car bodies. PhD thesis (2008)

[28] Arnal, L., Solanes, J.E., Molina, J., Tornero, J.: Detecting dings and dents on specular car body surfaces based on optical flow. Journal of Manufacturing Systems **45**, 306–321 (2017)

[29] Zhou, Q., Chen, R., Huang, B., Liu, C., Yu, J., Yu, X.: An automatic surface defect inspection system for automobiles using machine vision methods. Sensors **19**(3), 644 (2019)

[30] Park, S.H., Tjolleng, A., Chang, J., Cha, M., Park, J., Jung, K.: Detecting and localizing dents on vehicle bodies using region-based convolutional neural network. Applied Sciences **10**(4), 1250 (2020)

[31] Marchand, E., Uchiyama, H., Spindler, F.: Pose estimation for augmented reality: a hands-on survey. IEEE transactions on visualization and computer graphics **22**(12), 2633–2651 (2015)

[32] Belghit, H., Bellarbi, A., Zenati, N., Otmane, S.: Vision-based pose estimation for augmented reality: a comparison study. arXiv preprint arXiv:1806.09316 (2018)

[33] Thrun, S.: Simultaneous localization and mapping. In: Robotics and Cognitive Approaches to Spatial Mapping, pp. 13–41. Springer, New York, NY (2007)

[34] Nowacki, P., Woda, M.: Capabilities of arcore and arkit platforms for ar/vr applications. In: International Conference on Dependability and Complex Systems, pp. 358–370 (2019). Springer

[35] Chen, C., Zhu, H., Li, M., You, S.: A review of visual-inertial simultaneous localization and mapping from filtering-based and optimization-based perspectives. Robotics **7**(3), 45 (2018)

[36] Barnard, S.T., Fischler, M.A.: Computational stereo. ACM Computing Surveys (CSUR) **14**(4), 553–572 (1982)

[37] Murray, D., Little, J.J.: Using real-time stereo vision for mobile robot navigation. autonomous robots **8**(2), 161–171 (2000)

[38] Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision **47**(1), 7–42 (2002)

[39] Wang, Y., Lai, Z., Huang, G., Wang, B.H., Van Der Maaten, L., Campbell, M., Weinberger, K.Q.: Anytime stereo image depth estimation on mobile devices. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 5893–5900 (2019). IEEE

[40] Chen, Y., Tang, J., Jiang, C., Zhu, L., Lehtomäki, M., Kaartinen, H., Kaijaluoto, R., Wang, Y., Hyyppä, J., Hyyppä, H., *et al.*: The accuracy comparison of three simultaneous localization and mapping (slam)-based indoor mapping technologies. Sensors **18**(10), 3228 (2018)

[41] You, S., Neumann, U., Azuma, R.: Orientation tracking for outdoor augmented reality registration. IEEE Computer Graphics and Applications **19**(6), 36–42 (1999)

[42] Vassallo, R., Rankin, A., Chen, E.C., Peters, T.M.: Hologram stability evaluation for microsoft hololens. In: Medical Imaging 2017: Image Perception, Observer Performance, and Technology Assessment, vol. 10136, pp. 295–300 (2017). SPIE

[43] Scargill, T., Chen, J., Gorlatova, M.: Here to stay: Measuring hologram stability in markerless smartphone augmented reality. arXiv preprint arXiv:2109.14757 (2021)

[44] Schöps, T., Sattler, T., Häne, C., Pollefeys, M.: 3d modeling on the go: Interactive 3d reconstruction of large-scale scenes on mobile devices. 3DV **1**(2), 3–2 (2015)

[45] Brennan, P.F., Ponto, K., Casper, G., Tredinnick, R., Broecker, M.: Virtualizing living and working spaces: Proof of concept for a biomedical space-replication methodology. Journal of biomedical informatics **57**, 53–61 (2015)

[46] Ponto, K., Tredinnick, R., Casper, G.: Simulating the experience of home environments. In: 2017 International Conference on Virtual Rehabilitation (ICVR), pp. 1–9 (2017). IEEE

[47] Segal, A., Haehnel, D., Thrun, S.: Generalized-icp. In: Robotics: Science and Systems, vol. 2, p. 435 (2009). Seattle, WA

[48] Bylow, E., Sturm, J., Kerl, C., Kahl, F., Cremers, D.: Real-time camera tracking and 3d reconstruction using signed distance functions. In: Robotics: Science and Systems, vol. 2, p. 2 (2013)

[49] Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: 2011 10th IEEE International Symposium on Mixed and Augmented Reality, pp. 127–136 (2011). IEEE

[50] Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. ACM siggraph computer graphics **21**(4), 163–169 (1987)

[51] Ho, K.L., Newman, P.: Detecting loop closure with scene sequences. International journal of computer vision **74**(3), 261–286 (2007)

[52] Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M.: Real-time 3d reconstruction at scale using voxel hashing. ACM Transactions on Graphics (ToG) **32**(6), 1–11 (2013)

[53] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European Conference on Computer Vision, pp. 405–421 (2020). Springer

[54] Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7210–7219 (2021)

[55] Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10318–10327 (2021)

[56] Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Lassner, C., Theobalt, C.: Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12959–12970 (2021)

[57] Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: Nerf–: Neural radiance fields without known camera parameters. arXiv preprint arXiv:2102.07064 (2021)

[58] Goh, G.D., Sing, S.L., Yeong, W.Y.: A review on machine learning in 3d printing: applications, potential, and challenges. Artificial Intelligence Review **54**(1), 63–94 (2021)