Autoloader: Cargo Handling Software for Navy and Marine Aircraft

Jeremy Ludwig & Bart Presnell
Stottler Henke Associates, Inc.
San Mateo, CA 94402
ludwig, bpresnell @ stottlerhenke.com

Abstract— **Managing cargo loading for U.S. Navy and Marine Corps aircraft is a challenging task, requiring an understanding of elements such as aircraft limitations, aircraft center of gravity, cargo space dimensions, and tie-down procedures to name a few. These loading requirements specified in each aircraft's lengthy Cargo Loading Guide (CLG). To address the problem of efficiently and effectively stowing cargo, the U.S. Navy has proposed the development of an Android app that assists aircrew in completing their loadmaster duties.**

This paper describes the AutoLoader app, which performs calculations and provides feedback to help achieve efficient and effective cargo loading. Three specific capabilities are highlighted. The first creates 3D models for novel cargo using Augmented Reality. The second allows the user to develop scenarios that include 3D models of aircraft, cargo, and tie-down patterns, and then analyzes the tie-downs according to CLG-defined rules. The third uses genetic algorithms to automatically search for and efficient and effective tie-down patterns for a scenario. The primary contribution of this work is describing how existing tools from augmented reality, computer games, and artificial intelligence are brought together to rapidly prototype an end-to-end solution in this challenging domain.

TABLE OF CONTENTS

1. INTRODUCTION

Managing cargo loading for U.S. Navy and Marine Corps aircraft is a challenging task, requiring an understanding of elements such as aircraft limitations, aircraft center of gravity, cargo space dimensions, and tie-down procedures to name a few. These elements differ across aircraft and are documented in lengthy Cargo Loading Guides (CLGs). One solution to these challenges is developing an app that runs on an Android tablet and assists aircrew in completing their loadmaster duties, helping to ensure that cargo is stowed efficiently and meets loading requirements specified in the CLGs.

This paper describes the AutoLoader app, which performs calculations and provides feedback to help achieve efficient and effective cargo loading. The AutoLoader software has three primary requirements. First, enable the development of a 3D model of cargo placement and tie-down patterns. Second, evaluate the safety of the placement and tie-downs based on the information in the CLGs. Third, generate a complete solution, or finish a partial solution, to a specified problem.

AutoLoader includes three specific capabilities to achieve these requirements. The **AR Cargo Creator** creates 3D models of novel cargo using Augmented Reality (AR). The **3D Editor** allows the user to develop a 3D model of cargo and tie-down patterns, and then analyzes the tie-downs according to CLG-defined rules. The 3D Editor includes representations of multiple aircraft. The **GA Generator** uses genetic algorithms (GAs) to automatically generate efficient and effective tie-down patterns for a 3D model.

The following is an illustrative use case that ties these capabilities together. Using AutoLoader, a loadmaster or crew chief uses the 3D Editor to quickly create a 3D model that includes the aircraft configuration and cargo to be placed. In the less frequent case where one of the cargo items has not already been modeled, they will use the AR Cargo Creator to create a new cargo model. At this point, the user could either tie-down the cargo manually and press the 'Analyze' button to ensure the correctness of the solution or press the 'Generate' button to invoke the GA Generator and begin the search for a valid solution.

There are two main benefits to using AutoLoader, as opposed to the manual calculations that currently occur. First, there is a large number of constraints and issues that must be

978-1-7821-2734-7/20/$31.00 ©2022 IEEE

considered when securing a load of cargo. Doing these calculations manually, and ensuring none are missed, is a time-consuming task. By automating these calculations, AutoLoader will help loadmasters and crew chiefs to complete their work significantly faster than the current manual process. Second, the current system of manual calculation does not provide any feedback to inform the loadmaster of a possible mistake that might lead to a catastrophic event — either on the current flight or on a future flight if the mistake is repeated. AutoLoader provides specific, actionable feedback that helps the user correctly secure cargo in the current flight and provides loadmaster training through positive examples.

The primary contribution of this work is describing how existing tools from augmented reality, computer games, and artificial intelligence are brought together to rapidly prototype an end-to-end solution in this challenging domain. The Related Work section briefly describes the tools used in AutoLoader. Following that, the Methods and Results section describes what was built and analyzes the results. The Conclusion summarizes progress on AutoLoader to date and outlines future work.

2. Related Work

Augmented reality (AR) aims to provide an interactive experience the combines both real and virtual worlds [1]. One challenge of augmented reality is creating accurate 3D virtual models of real-world objects to interact with. The **AR Cargo Creator** component is tasked with just this problem - creating 3D models of novel real-world cargo. To do this, the AR Cargo Creator leverages the Unity Augmented Reality (AR) toolkit [2], which supports the development of immersive applications that interact with virtual and real-world objects. Specifically, we use the various features like plane tracking that are designed to capture 3D models. By using this toolkit AutoLoader builds on the massive amount of research that has gone into creating 3D models of real-word objects for AR systems. Snap2Cad [3] is a good example of ongoing research in this area related to AutoLoader. The Snap2Cad system uses information from an Android camera and the Google ARCore toolkit [4] to retrieve the most similar object from a library of 3D models, scale the model, and then visualize the combined scheme. AutoLoader will benefit as research such as this is incorporated in the AR toolkits.

Unity [5] provides a solid foundation for the **3D Editor**. Unity is an extremely popular real-time 3D development platform with extensive support for mobile devices. Unity supports the development of the features required by the user to place and restrain virtual cargo in a simulated aircraft on an Android tablet.

Genetic Algorithms (GAs) are a search technique inspired by the evolutionary process in biology [6]. The basic GA process is:

1. An initial population of solutions is created.
2. Offspring solutions are created through the process of combining two solutions through crossover and changing a solution through mutation.
3. A fitness function determines which offspring survive to create more offspring.

This process is continued until some threshold is reached such as number of generations. AutoLoader uses the open-source GeneticSharp [7] library as a foundation for the **GA Generator** that generates efficient and effective tie-down patterns.

Related to the analysis functionality of AutoLoader, we did not find any related work developing intuitive, tablet-based, software that models restraining cargo and analyzing the strength of the restraint. There is however significant work in analyzing aircraft weight and balance – a future feature of AutoLoader. For example, [8] describes a mixed integer programming model that maximizes payload and minimizes center of gravity deviation. The authors also provide an in-depth review of related work on air cargo weight and balance optimization. They also note that while there are existing visual software tools that calculate aircraft weight and balance many airlines still perform this task manually.

3. Methods

This section describes each of the three main capabilities included in the AutoLoader prototype. The system overview is shown in Figure 1 and presents the context for the three capabilities.

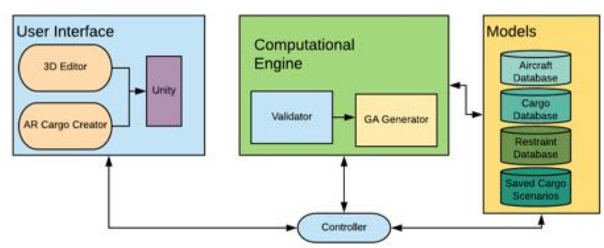

Figure 1. AutoLoader system overview.

User Interface

The **AR Cargo Creator** functionality leveraged the Unity AR Toolkit to allow the loadmaster to interactively create a 3D model from a real-world object, add tie-down points to the model, and to import the model into the 3D editor app as shown in Figure 4.

Testing the AR Cargo Creator, we found good accuracy with the picture-based plane detection, where known images are placed on each plane, on the order of 2–5cm of error in each direction for a 2m x 2m x 2m cargo box. The system worked under a wide range of angles, with the limiting factor that the picture must fill about 50% of the frame. Finally, the system continued to work under very low light (down to 1 lux).

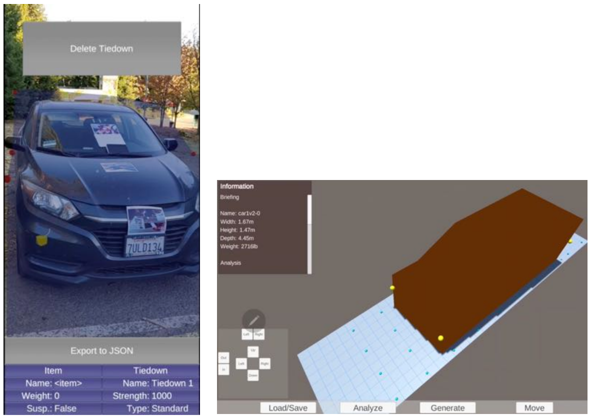

Figure 4. Creating a Model of a Car in AR Cargo Creator (left), Importing the Model into the 3D Editor (right).

The **3D Editor** leverages the Unity Engine to visualize cargo placement and tie-down patterns. The 3D Editor is also responsible for displaying the results of the Validator, which include the restraint in each direction and that specific restraint rules are followed. Figure 2 (left) shows an example of sufficiently restrained cargo in a heavy lift helicopter. The green arrows indicate sufficient restraint in all directions. The lack of a warning icon above an object indicates that all restraint rules are followed. Figure 2 (right) illustrates an incorrectly restrained item in another aircraft, where the cargo box on the right has both insufficient restraint and rule violations. The cargo box on the left is properly tied down. Finally, the yellow blocks on the far left indicate unavailable/unusable tie-down locations.

Computational Engine

Initial versions of both the Validator and **GA Generator** were developed and demonstrated. Once the cargo is placed and tied-down in the 3D Editor, the Validator analyzes the cargo restraint provided by tie-downs, utilizing the formulas outlined in the CLGs. Further, the editor validates that specific restraint rules are followed such as: do not use blocked tie-down points; if a cargo object has suspension, then 50% of the restraint must be above the suspension; the fore/aft and port/starboard restraints must be balanced; and mismatched restraints (e.g., chains and straps) should not be used.

Within the 3D Editor app, the 'Generate' button launches the **GA Generator**. This capability uses a genetic algorithm to automatically search for an efficient and effective tie-down pattern for a 3D model, as shown in Figure 3. In the GA

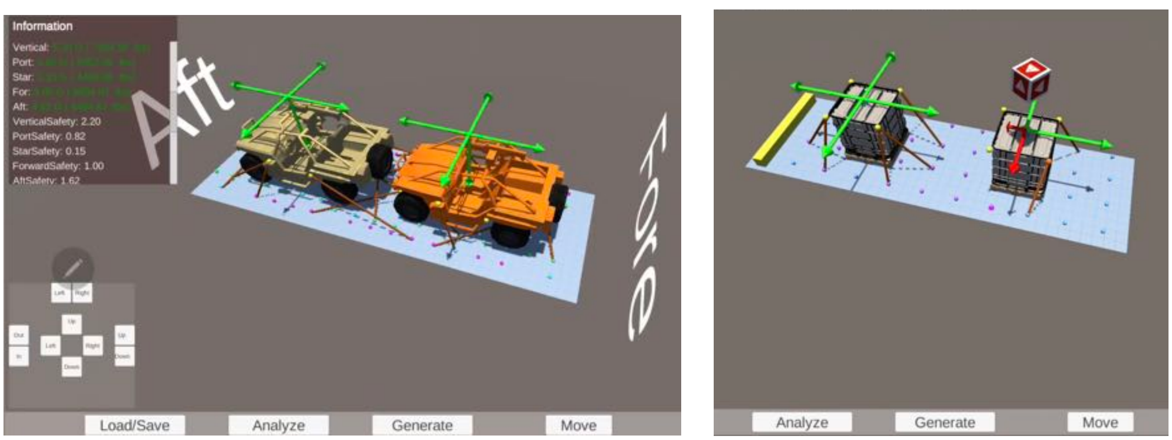

Figure 2. Sufficiently restrained (left) and insufficiently restrained (right) cargo.

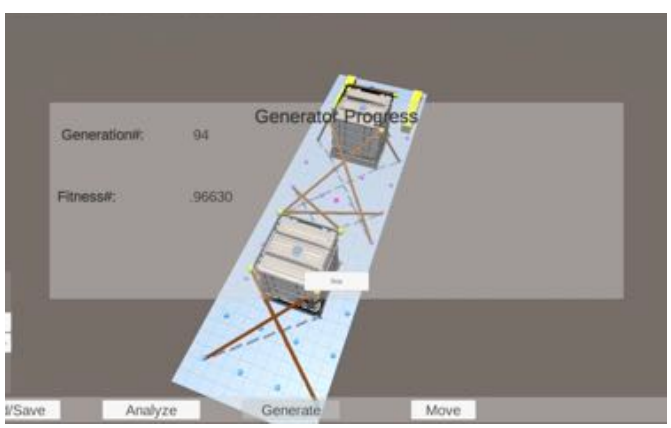

Figure 3. GA Generator in progress.

representation, each chromosome is a solution. The chromosome is made up of an ordered list of genes, where each gene is a cargo item and its restraints. The crossover function combines the genes (restrained cargo items) of one parent with that of another to create a new solution (Figure 5). The mutation function will, with a small probability, randomly modify a gene. Pragmatically, this mean adding, removing, or editing a restraint. The fitness function drives the search, where only the best solutions survive to populate future generations. The fitness function improves and refines results over time by ensuring the required restraint is met in all directions, minimizing the number of rule violations, preferring fewer and shorter straps, and encouraging symmetry.

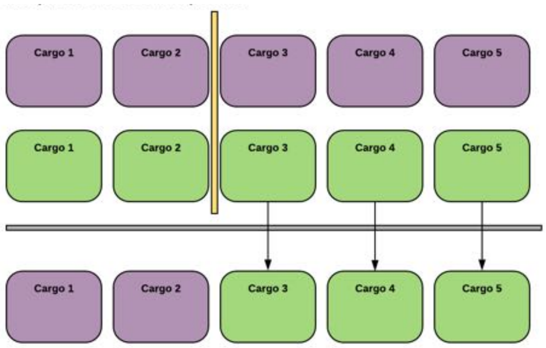

Figure 5. Genetic algorithm representation.

Models

A set of Aircraft and Restraint models were developed for the prototype app based on the information in the CLGs. For the Pre-existing 3D models were used for the cargo items. collected for the prototype app.

4. RESULTS AND DISCUSSION

The prototype app demonstrated the technical feasibility of all three capabilities on a Samsung Galaxy Note 10+. The AR Cargo Creator proof of concept was demonstrated by capturing a 3D model for a Honda HRV in outdoor lighting and on uneven ground. We found reasonable accuracy using AR to capture planes in the real-world tests, though the models required some simple, interactive editing during capture. The AR Cargo creator includes support for this type of editing, and for adding tie-down points that corresponded to the vehicles tow hook locations. The resulting models can be imported into the 3D Editor.

The 3D Editor proof of concept was demonstrated by placing and tying down cargo items in two different aircraft. The 3D Editor also displayed the results of the Validator, effectively communicating the restraint provided in each direction and that specific restraint rules are followed. Notably, all of this is being done on a relatively small tablet screen.

Finally, the GA Generator proof of concept was demonstrated by using a genetic algorithm to search for tie-down solutions in multiple aircraft/cargo scenarios. Running inside the 3D Editor, results obviously improve over time as the fitness function continuously improves results over time. The end results are judged to be reasonable by subject matter experts, though with plenty of room for improvement. The GA Generator takes 2-3 minutes to find a reasonable solution within the limited processing power of the Samsung Galaxy Note 10+.

5. CONCLUSION

Managing cargo loading for U.S. Navy and Marine Corps aircraft is a challenging task, requiring an understanding of complex requirements that differ across aircraft and are documented in lengthy Cargo Loading Guides (CLGs). The primary objective of the AutoLoader system described in this paper is to assist aircrew in completing their loadmaster duties — helping to ensure that cargo is stowed efficiently and meets loading requirements specified in the CLGs.

Using the **AR Cargo Creator**, we demonstrated the capability to develop an initial model using augmented reality to recognize planes, to manually refine the model and add tie-down points, and to import the created model into the 3D Editor. Using the **3D Editor**, we demonstrated placing and tying down cargo in multiple aircraft. The engine quantified the restraint applied to cargo in each direction, validated that restraint rules were followed, and the editor communicated this information to the user. Using the **GA Generator**, we demonstrated the ability to use genetic algorithms to efficiently search for a tie-down solution. The fitness function improved and refined results over time by ensuring the required restraint was met in all directions, minimizing the number of rule violations, preferring fewer and shorter straps, and encouraging symmetry. The prototypes successfully demonstrate the technical feasibility of key aspects of AutoLoader.

While the protype results successfully demonstrate the technical feasibility of key aspects of AutoLoader, more work is needed to move from the prototype to a minimum viable product. This work includes improving the usability of the tool and adding features based on end-user use cases. For the AR Cargo Creator, these include capturing complex objects like cylinders and stacks of boxes, as well as tracking emerging phone/tablet features such as LIDAR that might improve AR results. For the 3D Editor, these include performing center of gravity calculations, handling a wider range of cargo models, handling cargo with no tie-down points, and automatically applying standard tie-downs for known cargo. Finally, for the GA Generator these include exploring genetic algorithm parameters, seeding the initial population with standard tie-downs, refining the fitness function, and optimizing the placement of cargo in addition to tie-downs. These changes would both increase the speed of the GA Generator and improve the quality of results.

ACKNOWLEDGEMENTS

This material is based upon work supported by the United States Navy under Contract No. N68335-20-C-0538. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the United States Navy. NAVAIR Public Release 2021-549 Distribution Statement A - "Approved for public release; distribution is unlimited".

REFERENCES

[1] "Augmented reality," *Wikipedia*. Dec. 30, 2021. Accessed: Dec. 30, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Augmented_reality&oldid=1062792894

[2] U. Technologies, "Augmented Reality Development Software | AR Engine for Apps | Unity." https://unity.com/unity/features/ar (accessed Nov. 11, 2020).

[3] A. Manni, D. Oriti, A. Sanna, F. De Pace, and F. Manuri, "Snap2cad: 3D indoor environment reconstruction for AR/VR applications using a smartphone device," *Computers & Graphics*, vol. 100, pp. 116–124, Nov. 2021, doi: 10.1016/j.cag.2021.07.014.

[4] "Build new augmented reality experiences that seamlessly blend the digital and physical worlds | ARCore," *Google Developers*. https://developers.google.com/ar (accessed Dec. 30, 2021).

[5] U. Technologies, "Unity Real-Time Development Platform | 3D, 2D VR & AR Engine." https://unity.com/ (accessed Nov. 11, 2020).

[6] "Genetic algorithm," *Wikipedia*. Nov. 11, 2020. Accessed: Nov. 18, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Genetic_algorithm&oldid=988134173

[7] "GitHub - giacomelli/GeneticSharp: GeneticSharp is a fast, extensible, multi-platform and multithreading C# Genetic Algorithm library that simplifies the development of applications using Genetic Algorithms (GAs)." https://github.com/giacomelli/GeneticSharp (accessed Nov. 11, 2020).

[8] X. Zhao, Y. Yuan, Y. Dong, and R. Zhao, "Optimization approach to the aircraft weight and balance problem with the centre of gravity envelope constraints," *IET Intelligent Transport Systems*, vol. 15, no. 10, pp. 1269–1286, 2021, doi: 10.1049/itr2.12096.

BIOGRAPHY

Jeremy Ludwig, *PhD, is a principal engineer at Stottler Henke Associates, Inc. He directs teams of computer scientists and conducts research in artificial intelligence, applying reasoning, knowledge representation, and machine learning to create solutions for complex, real-world, problems.*

Bart Presnell *is a software engineer at Stottler Henke Associates. His past work includes technical lead for several intelligent tutoring and game-based training systems as well as for numerous planning and scheduling projects.*