Utilizing Augmented Reality to Update Nautical Charts

Tyler Kokoszka, Harrison Pham, Briana Sullivan, and Thomas Butkiewicz
Center for Coastal and Ocean Mapping
University of New Hampshire
Durham, New Hampshire, USA

***Abstract*— Many mariners, especially recreational boaters, still utilize paper nautical charts for navigation. Unlike electronic charts, regularly updating these paper charts with new information can be a tedious task. Changes to paper charts are published in a textual format, which mariners must then use to manually locate on their physical chart by using a compass, pencil, and ruler to carefully place the update in the proper position on the chart. This project investigates the potential for using augmented reality (AR) to simplify and expedite the updating process. AR is a technology that superimposes digital information directly on top of a user's real world view. This project uses off-the-shelf, self-contained AR glasses (Microsoft HoloLens) to allow mariners to look at their paper nautical charts and see all modifications that need to be rectified and their respective locations on the paper chart. Advances in both AR and smart-phone technologies imply that this application could be implemented as a mobile app in the near future, which would make it easily accessible to average mariners.**

Keywords— augmented reality; charting; navigation

I. Introduction

Mariners of all kinds use nautical charts to navigate through bodies of water. Nautical charts are available in both paper and electronic formats. Constant changes in the physical environment need to be reflected with changes to these charts. Electronic charts are typically updated automatically via online mechanisms. Paper charts, however, must be updated manually, using physical tools and a publication from the United States Coast Guard (USCG).

Each week the USCG releases chart updates, called the Local Notice to Mariners. These updates consist of a list of textual information for each chart (e.g. update name, update type, longitude, latitude, etc.) and can be found in a .pdf document hosted on the USCG website. The process of updating a chart using these textual descriptions can be tedious, time consuming, and error prone. While this process requires only basic map skills, for charts with 100+ updates, it can take a significant amount of time, or an undesired monetary cost to print/purchase a new chart.

A web application called Chart Update Mashup (ChUM) was created by Sullivan[1] as a first step to help aid the paper chart update process. ChUM utilizes the Google Maps API by displaying a digitized nautical chart at the corresponding real world location, and overlays all the chart updates on it. This allows users to quickly see where each of the updates are on the chart. The main drawback to this solution is that the user still has to look back and forth between the web application and their paper chart. The second step in her research was to introduce a more intuitive approach by overlaying the updates directly onto the physical chart. Sullivan attempted to use an android phone, its camera, and image processing to create the augmented reality (AR) solution. However, the technology at the time wasn't sufficiently advanced to realize the goal.

This project focused on developing an augmented reality version of ChUM using the Unity engine for deployment on Microsoft's HoloLens AR device. The application receives a chart number and chart bounds as input from the user. It then pulls updates of the respective chart from NOAA's website, uses the bounds of the paper chart to calculate the placement of each update, and projects 3D representations of the updates onto the user's view of the paper chart. Then, a pen can be used to mark the updates on the chart where the AR icons appear. Fig. 1 shows an example view of what this presentation looks like to the user.

Fig. 1. User's view, showing virtual update markers overlaid on a real-world, physical chart.

II. Background

For a mariner to update a chart, they first need to download the latest corresponding Local Notice to Mariners (LNM) document from the USCG website.[2] Within these LNM .pdf documents, updates are listed in both paragraph and tabular formats, divided into various sections including special notes,

This project is supported under NOAA grant #NA15NOS4000200.

discrepancies, temporary changes, chart corrections, proposed changes and advance notices. This project focuses only on the "chart corrections" section of the LNM that can also be found on the Office of Coast Survey's (OCS) website.[3] By inputting the chart number on the OCS website, chart corrections are returned in a tabular format that can be downloaded as a text file.

With the tabular form of the chart corrections, the mariner then needs to check the “last update date” on the bottom of their chart to determine which edition of the notices will contain all the necessary changes. The data available within the table of chart corrections includes: Action, Item Name, Charting Label, Latitude, Longitude, Published Document, Kapp, RNC Panel, RNC Posted, PDF Posted.

The Action consists of: add, delete, relocate, change, substitute. This determines what kind of change is taking place with the item in question. The item name is a description of the charting label, e.g. a number to represent the depth at a location, letters and numbers to represent a navigational aid marker and it’s characteristics, or a name of a feature. These descriptions might also include other attributes describing the feature, e.g. the clearance height for a bridge. The latitude and longitude gives the real world location of the item, and the other fields contain metadata. This project focuses on the non-metadata related data.

III. Technologies

This prototype required AR hardware to superimpose virtual objects on the real world, and software to enable interaction with the virtual objects. The follow technologies were used together to create the final prototype.

A. *Microsoft HoloLens*

The Microsoft HoloLens[4][5], an off-the-shelf, self-contained AR headset, was chosen for this project because it was the first and only mass-produced consumer AR device available at the time. Because it is a lightweight wearable computer, it has relatively low computing specs (2GB RAM, 64GB flash memory), which unfortunately limited the complexity of this application.

HoloLens uses time-of-flight depth cameras to create 3D maps of surrounding surfaces. This allows virtual objects to be anchored to, and collide with, surfaces; producing the effect that virtual objects are located in the real world. A gaze-directed cursor can be projected in 3D onto whatever surface/object is at the center of the user’s view. Navigation and interaction can be done via hand gestures (e.g. tapping or “clicking” with the index finger) and through speech recognition.

B. *Unity*

The software for this project was created using the Unity game engine.[6] Unity provides a beginner-friendly interface for developing interactive 3D games and simulations, and allows for cross-platform integration with many technologies (e.g. augmented and virtual reality systems, WebGL, etc.) Unity was an attractive solution for this project because, in addition to supporting the HoloLens interface, it provides most of the required underlying functionality and algorithms (e.g. rendering, physics, GUI elements). The custom Unity scripts created for this project were written in C#.

C. *Vuforia*

Vuforia[7] is an image recognition tool that (as of Unity v2017.2) is packaged as an optional tool when installing Unity. Vuforia can recognize predefined images based on precomputed visual features. For an image to be recognized by Vuforia, it must be uploaded to Vuforia’s website for processing. A Unity package is then available for download, which can be imported into a Unity application to enable the image to be recognized in a Unity scene. Approximate real world size, in meters, and any additional functionality is specified within Unity.

IV. Methodology

A. *Chart Calibration*

The most challenging aspect of this project was calibrating the system such that the virtual content is correctly aligned with the physical chart. While the HoloLens is able to detect the 3D surface of the user’s chart, it is unable to identify the orientation of the chart or the edges between the chart and the table it rests on.

Initial attempts focused on using image recognition to detect and locate the chart in the camera’s view. It was hoped that views of the physical chart from the HoloLens camera could be compared to a known high-resolution image of the chart, and used to calculate the location and extents of the physical chart. However this approach was unsuccessful, primarily due to the size and complexity of nautical charts. Charts are very large (approximately 30 inches x 40 inches), and contain many small details (e.g. numbers, letters). Feature-based image recognition would have to be done at multiple levels of detail simultaneously, because small, high-frequency details such as depth sounding numbers are ambiguous due to being repeated many times across the chart, and would only be resolvable when the camera is close to the chart, while large, low-frequency details such as shaded land masses and shorelines would only be detectable when the camera is far enough away from the chart to see most of it at once. This effect can be seen in Fig. 2, where the same part of a chart is viewed from increasing distances. Attempting to use whole-chart images with Vuforia confirmed that these challenges made automatic, image recognition based calibration impractical.

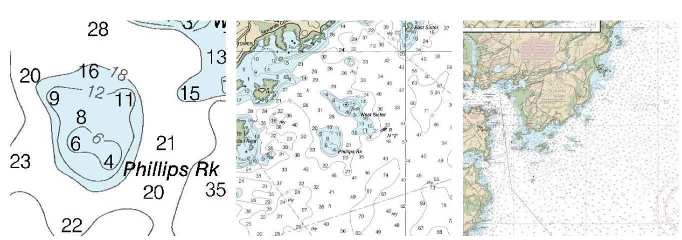

Fig. 2. The same location on a paper chart, viewed at increasing distances. Note that different features are recognizable at different distances.

Instead, an interactive manual calibration method was devised. To calibrate a chart, the user gazes at each corner of the physical chart, and when the cursor is over the corner, they perform a tap gesture. This places a virtual anchor point at that location. This process yields 4 corner placements that represent the bounding box of the physical chart in the virtual space. This worked, but was prone to inaccuracies. User error when selecting the corners often resulted in projection distortion and displacement.

To simplify the process and increase accuracy, a hybrid combination of the two previous solutions was implemented using AR markers. AR markers (as shown in Fig. 3) are small, printed images that are designed with features that are easily recognized by computer vision algorithms such as Vuforia. Our prototype has the user place these small AR markers at the corners of their chart, where they are readily detected by Vuforia. This more consistently generates accurate real-world coordinates for the virtual chart anchors, and is generally faster than the completely manual calibration solution.

Fig. 3. A paper AR marker placed over a chart corner, with virtual anchor (white cube) that confirms it has been correctly recognized and located.

A similar strategy, wherein just the corners of the full-resolution chart images are used in place of AR markers may be possible. This was not attempted here, however, because of the excessive time-costs of cropping and uploading the corners of thousands of charts to Vuforia for processing.

Testing revealed it was possible to embed the AR markers directly onto the corners of digital charts before printing them, as long as the markers' real world printed size was consistent (i.e. the map was printed at 100% scale). An example of this is shown in Fig. 4. Likewise, a QR-code could be embedded that contains the chart number, size, and print date.

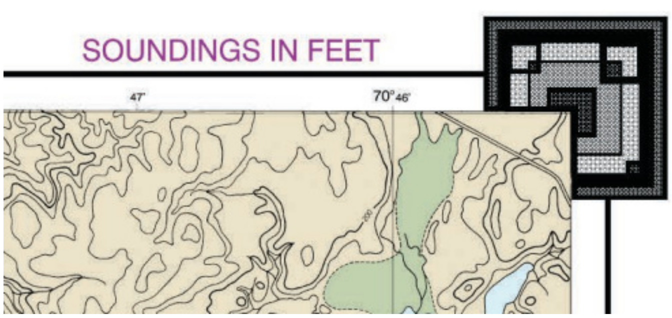

Fig. 4. An AR marker embedded directly into a chart file before printing.

B. Projection of Updates

Projecting the updates onto a virtual map overlay requires manipulation through linear algebra transformations. Locations for each update are given in lat/long coordinates through NOAA's Notice to Mariners text file. These coordinates are converted to virtual map coordinates through a window-to-viewport transformation. The transformation involves translating the window to the origin, scaling the window to the size of the viewport, and translating the window to the viewport; where the window is the chart and the viewport is the virtual map in the Unity environment.

Lat/long 2D coordinates were projected to 3D real world coordinates by ray casting from a 2D plane onto the 3D triangulated mesh of the physical chart's surface. The end result can be seen in Fig. 5, where virtual update markers appear to float directly above the appropriate locations on the real chart.

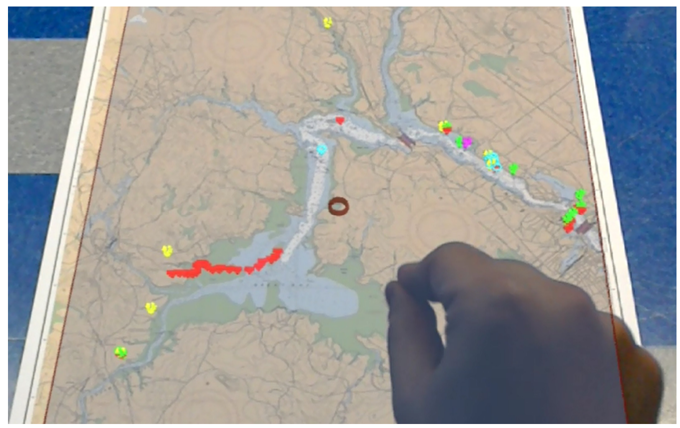

Fig. 5. User's view, showing color-coded 3D representations of chart updates overlayed onto a paper chart.

C. Gestures & UI

Hand gestures (primarily the "tap" or "finger click" gesture) are used to interact with various objects/holograms via the use of raycasting through the gaze cursor and collision detection. This solution was preferable because it provides a "point-and-click" or "action/reaction" experience for the user. The user looks at an object to align the cursor, and "clicks" on it; an intuitive and familiar interaction experience with a minimal learning curve.

A significant drawback was difficulty interacting with small objects. Trying to focus the small cursor on a small object can be difficult due to "jitter" caused by micromovements of the user and sensor inaccuracies. Additionally, if many small objects are clustered together in one area, it can be hard to select a specific object.

To keep the chart area clear, GUI elements were placed on a virtual heads up display (HUD), which utilizes the empty 3D space above the chart/tabletop. It contains UI buttons for performing various interactions with the chart updates and the chart itself. It was found that providing easy to click buttons on the HUD was preferable to having to remember many gestures for the various functions. Small object interactions were facilitated by zoomed-in map views within the HUD.

V. FEATURES

A. Chart Input

The Chart Input is a virtual keypad (shown in Fig. 6) that allows the user to specify which nautical chart they wish to view updates for. It can work online or offline. When online, the user enters a chart ID number, and the application pulls that chart's updates, and any information related to it (e.g. Google Polylines information) from NOAA's website. Alternatively, when offline, it pulls the information from a locally stored copy that was previously downloaded. The HoloLens performs best in offline mode, as it takes a long time to download and process the data, yielding a poor user experience.

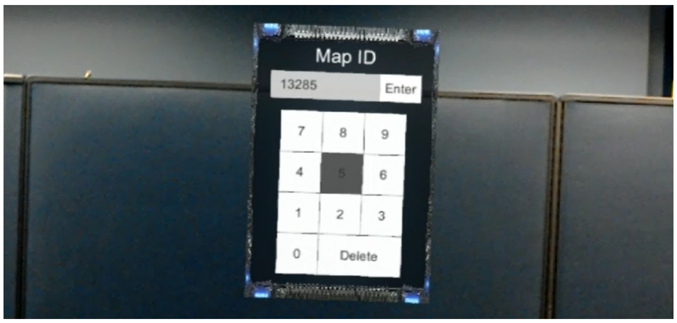

Fig. 6. Keypad to specify which NOAA chart number to retrieve updates for.

B. Virtual Map Overlay

The Virtual Map Overlay encapsulates the representations of the chart updates over the physical chart. Individual updates are represented by a 3D model that hovers on top of the physical map. Each model corresponds to a update type; delete, add, relocation, substitution, or change.

Due to performance and visibility concerns, two versions of update icons were created; detailed and simple. Simple models are a thin, vertical cylinder that is color coded to indicate update type. Detailed models include a symbol that redundantly represents the update type. To minimize hologram overlap, update models are scaled to approximately 1 cm tall and 0.5 cm wide.

C. Virtual HUD

A virtual heads-up-display (HUD), as shown in Fig 7, provides a central interface for data manipulation and customization of the application. It provides a legend, more detailed information about updates, and tools for filtering and changing views.

Fig. 7. Virtual HUD for interaction and data manipulation.

The topological view acts as an amplified top-down perspective of the Virtual Map Overlay. A Unity camera was placed above, and aimed at, the center of the map. Although the image rendered on the HUD only includes the virtual scene (i.e. no live camera feed), it provides the user another perspective on a desired area. Controls are provided to zoom, rotate, translate, and reset the camera's view.

A filtering interface helps users minimize clutter and prevent information overload. Updates can be filtered based on update type, so that users can isolate which updates to view. Although filtering by type is helpful, being able to filtering further, down to individual updates, was essential. A selectable update point list was implemented to allow specific updates to be selected. Each item in the list displays a checkbox for toggling, polar coordinates, update type, and a description. However, due to hardware limitations, the application encounters significant latency when generating this list. To counteract this, a button was added to the Virtual HUD to toggle the list as needed.

D. Selection Zoom

The Selection Zoom tool provides a magnified view of an area on the Virtual Overlay. By pinching two fingers together, and holding them together, a pop-up window appears above the gaze cursor. This pop-up window displays a magnified view of a small area of the virtual overlay around the user's cursor. This is helpful in areas where updates a clustered close together, and allows the user to see individual updates more clearly when marking those updates on the physical chart.

VI. RESULTS

This proof-of-concept prototype accomplished the primary goal of accurately displaying chart updates on a physical chart utilizing augmented reality. Basic functionality is provided for manipulating the presented information and facilitating the chart update process. Through the use of AR markers, the time required for calibration was reduced to under one minute. The application removes the need for measurement and plotting tools, and requires only a pen or pencil.

The prototype was tested in-situ on board The Center for Coastal and Ocean Mapping's RV Gulf Surveyor while it was docked. The application worked at first, but yielded inaccurate results after the first attempt. After further investigation, the issue was related to the initial position and orientation of the HoloLens and the physical chart when starting the application. Previous tests had the HoloLens facing the chart vertically, producing the desired results. However, due to limited space on the boat, the HoloLens was required to face the chart horizontally, resulting in inaccurate update placements.

There was concern about potential conflict/mismatch between the HoloLens's motion sensors and visual tracking system when on a moving vessel. As the boat rocked in the waves, the motion sensors would report the device moving, but the visual tracking system would report the device as not moving (relative to the rest of the cabin's interior). However, aside from some static inaccuracies, the virtual updates rarely moved as the boat swayed. It is assumed that this issue can be largely avoided by disabling the motion sensors and relying only on visual tracking when the application is to be used on a moving vessel.

An informal evaluation was conducted on board with the research vessel's captain. The captain was fitted with the HoloLens and shown how to use the application to update a chart on the boat's chart table, as shown in Fig. 8. He was impressed by the technology and noted how the application could be helpful. He expressed how the traditional approach of compiling all of the chart updates was difficult and cumbersome. He estimated it takes about an hour to update a single chart, and thus he only updates his charts about once a year. Because the application would handle the compilation of the updates and allow him to update a chart within minutes, he expected he would be able to update his charts more frequently.

Fig. 8. Team members help guide the RV Gulf Surveyor's captain in using the HoloLens and AR ChUM application.

VII. LIMITATIONS

Hardware limitations of the HoloLens device presented some challenges that held back development of our prototype. Because it is self-contained, the hardware is not very powerful in terms of computational performance; a necessary trade-off in order to keep the device small and lightweight enough to be wearable on one's head. This limited the number of updates able to be displayed at once. A competing AR product, the Magic Leap One[8], uses a belt-mounted computer, which allows for larger and heavier processing hardware without encumbering the user's head.

The field-of-view of the HoloLens is extremely narrow, approximately 35 degrees diagonally. This means that the virtual chart overlays are only visible in the very center of the user's vision. The Virtual HUD, for example, cannot be seen at the same time as the user is looking downwards at their chart. Instead of glancing between the HUD and chart, the user has to physically move their head back and forth. While downsizing the interface elements, and placing them closer together can somewhat alleviate these issues, it results mostly in clutter and occlusion issues. The next version of HoloLens, expected in 2019, is rumored to have twice the field-of-view, which would be a most welcome improvement.

While our manual chart calibration procedure works well, it cannot account for the natural tendency for a user to move their chart around on the table while editing. This issue could be alleviated by a computer vision based calibration method that automatically recognizes and tracks the chart, but this would be challenging to implement robustly. It is likely that future AR software development frameworks will soon provide improved image recognition and tracking functionalities that can handle this task satisfactorily.

Cost is a significant factor limiting transition of this application from proof-of-concept to practical product. HoloLens costs approximately $3,000 USD. Although this application may yield a desirable experience for the user, the price of the hardware makes it impractical for the average consumer. It is expected that, like most consumer computing devices, future products will offer more functionality at lower costs. Furthermore, as smartphone manufacturers continue to incorporate AR technology into their phone's, eventually this type of application will be able to be deployed as an app on average mariners' existing devices.

VIII. CONCLUSION

Currently, mariners who want to update their physical charts have to follow an inefficient workflow, which includes going to NOAA's website, finding their chart(s) on the Notice to Mariners, and then for each update, they must find the latitude and longitude coordinate on their physical chart, and mark it up. This process can be tedious and time-consuming, which results in most mariners updating their charts infrequently, if ever.

The proof-of-concept AR ChUM application presented here provides a solution to this problem, by speeding up and aiding in the updating process. It automatically retrieves the necessary updates from NOAA, and uses AR technology to display them in the form of a virtual overlay placed directly on the user's physical chart. This removes the repetitive, back-and-forth process of manually locating each update, and reduces the chances of making errors.

While this project succeeded in its goal of demonstrating that augmented reality has great potential to aid in updating of paper nautical charts, it was constrained by the significant limitations of the first-generation AR hardware available.

Ultimately, as augmented reality technology advances, costs will come down, and capabilities will expand. Eventually, improved software and the integration of spatial sensors into smartphones will put AR technology into the hands of average mariners. At which point, they will be better able to keep their charts updated, and thus enjoy improved safety of navigation.

ACKNOWLEDGMENT

This project is supported under NOAA grant #NA15NOS4000200.

REFERENCES

[1] B. M. Sullivan, “ChUM: Chart Update Mashup”, IEEE OCEANS ‘12. IEEE/MTS, Hampton Roads, VA, 2012.

[2] United States Coast Guard, “Local Notice to Mariners”, ‘https://www.navcen.uscg.gov/?pageName=lnmMain’

[3] Office of Coast Survey, “Notice to Mariners Updates”, ‘https://ocsdata.ncd.noaa.gov/ntm’

[4] Microsoft, “Microsoft Hololens”, ‘https://www.microsoft.com/en-us/hololens’

[5] Microsoft, “Hololens Hardware Details”, ‘https://docs.microsoft.com/en-us/windows/mixed-reality/hololens-hardware-details’

[6] Unity Technologies, “Unity 3d”, ‘https://unity3d.com/’

[7] PTC Inc., “Vuforia Sdk”, ‘https://developer.vuforia.com/downloads/sdk’

[8] Magic Leap Inc, “Magic Leap Creator Edition”, ‘https://www.magicleap.com/Magic-Leap-One’