Technical Disclosure Commons

Defensive Publications Series

October 05, 2017

DETERMINATION OF DEVICE POSE USING TEXT MATCHING FROM CAPTURED IMAGES

Tilman Reinhardt

Jeremy Pack

Allen Hutchison

Daniel Filip

Brian Brown

DETERMINATION OF DEVICE POSE USING TEXT MATCHING FROM CAPTURED IMAGES

ABSTRACT:

Both server-based and client-based systems can be used for precisely detecting orientation of a user device. Existing sets of geographically referenced imagery include identifiable text, and the same text may appear within multiple images within the set where the multiple images are taken from different viewpoints and camera angles. Multiple observations of the same physical text in the world are used to triangulate measurements and create a 3-dimensional (3D) location and direction for a specific text. The 3D location for the specific text is used to create a spatially indexed 3D text database, which can be located at a server and/or downloaded onto the client device. As a client device captures an image, character recognition is performed on the image, and the recognized text is compared to the 3D text database to find database text that matches the recognized text in the image. Once one or more text pieces have been matched, a triangulation computation can derive the device's orientation and location.

BACKGROUND:

A user device, such as a smartphone, in a dense urban environment has only a rough estimate of the location and orientation of device. This is typically derived from the sensors on the phone and GPS signals. For high precision augmented reality applications, such as overlaying geolocated content into the camera view, the orientation of the device needs to be precise to <1° and its location to 1m.

DETAILS:

A precise 6-degree-of-freedom pose of a client device can be determined by comparing text in images captured by the client device with a 3-dimensional (3D) geo-located text database. The 6 degree-of-freedom pose may include 3 location parameters, such as x, y, z, coordinates, and 3 orientation parameters, such as yaw, pitch, and roll.

As the client device captures an image, character recognition is performed on the image, and the recognized text is compared to the 3D text database to find database text that matches the recognized text in the image. Once one or more text pieces have been matched, a triangulation computation can derive the device's orientation and location.

The 3D text database may be created from a large data set of geo-referenced imagery. From this set, text can be identified using optical character recognition (OCR) algorithms. As there are multiple observations of the same physical text in the world it is possible to triangulate the measurements and create a 3D location and direction for a specific text. Accordingly, each specific text may be stored in association with a 3D location and direction. In its simplest form, the 3D text database is very small, with each record storing the text, xyz text pose, and an error ellipsoid. This allows the client device to load the 3D text database into memory and do the localization on the device without requiring a server connection. In other examples, the 3D text database may also store a direction normal to a plane of the text, an extent of the text (e.g., a bounding box), and other attributes, such as foreground, background, color, font, case (e.g., upper/lower case), etc. The 3D text database may be preexisting or created upfront using known information and updated from time to time. In other examples, the database may be created over time through repeated observations from coarsely localized devices which start adding 3D text to the database, starting with very large error ellipsoids, and refining the text location over time.

The client device can be any type of mobile electronic device capable of capturing and processing images. For example, the client device may be a smart phone, tablet, video game system, or the like. In some examples, the 3D text database is stored in a memory of the client device. For example, the client device may communicate with a server over a wired or wireless network, and download the 3D text database from the server. In other examples, the 3D text database may be loaded onto the client device through a storage medium such as a flash drive, USB memory device, etc. Regardless of how the 3D text database is loaded onto the client device, it may also be periodically updated.

The client device captures an image of its surroundings. For example, the client device may be turned to camera mode, video mode, to an application, or to any other mode in which images are received through a camera or other image capture device of the client device. Information, specifically text, from the images is detected, such as through optical character recognition. Detected text in the image is compared to the 3D text database. For privacy reasons, the text could in some examples be encrypted with a one-way algorithm, such as a hash. When matches are found for the detected text, location information from the 3D text database is associated with the detected text. Triangulation techniques are then used to determine a pose of the client device. The pose includes orientation in x, y, and z directions, as well as location in x, y, and z directions.

Fig. 1 below illustrates an example image captured by an image capture device of a client device. The image includes three pieces of text. Those three pieces of text are encircled in Fig. 1 in order to highlight them, but will not necessarily be encircled in the captured image or in later processing of the captured image. The three pieces of text may be recognized using, for example, optical character recognition. Once recognized, they are compared to the 3D text database. Matching text in the 3D text database may have associated location information.

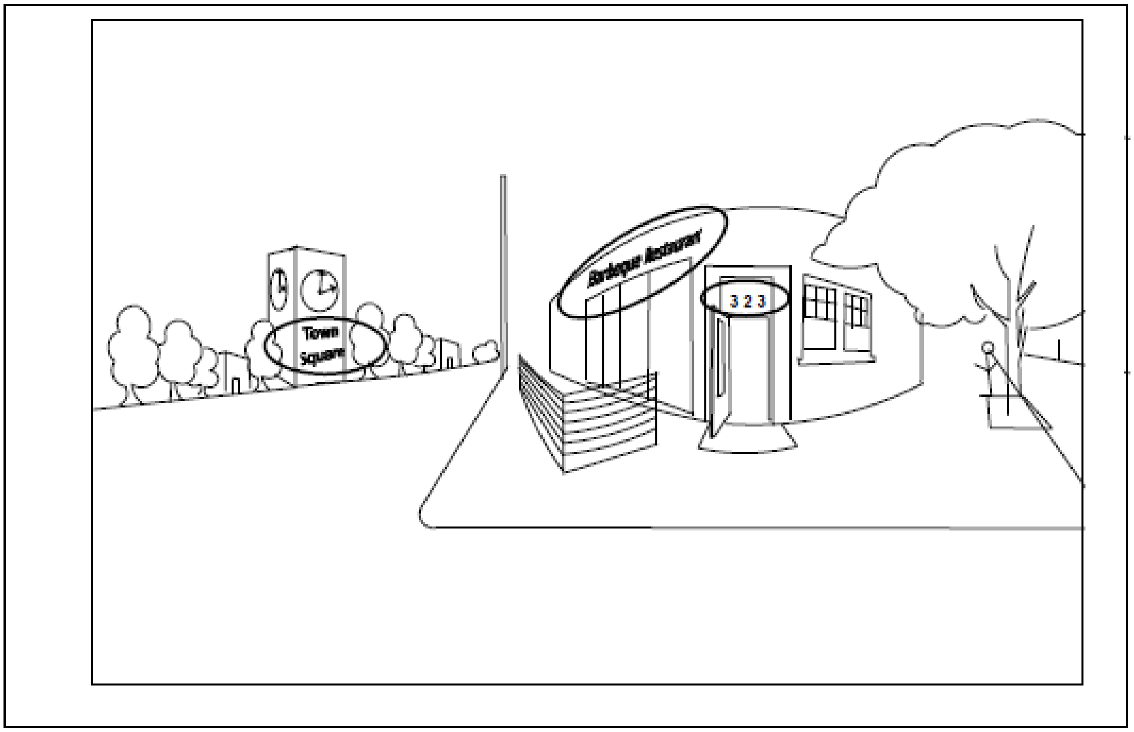

Fig. 1 – Example image captured by client device, with detected text highlighted.

The location information associated with the text matches in the 3D text database includes direction information as well as positional information. For example, an x, y, z coordinate may be associated with each word, with each group of words, or with each letter. Additionally, a direction in which each word or group of words reads from left to right is also associated with each word or group of words. For example, as shown in Fig. 2 below, a first piece of text "Town Square" extends in a first direction, a second piece of text "Barbeque Restaurant" extends in a second direction, and a third piece of text "323" extends in a third direction. In this example, each of the first, second, and third directions are different, though in other examples directional information for two or more pieces of text may be the same. The directional information may also be defined along an x,y,z axis, using vectors, or by any other mechanism for indicating direction.

According to some examples, directional information of the detected text may also be determined from the captured image. For example, vector analysis may be used to determine a direction for the detected text, and use that information to determine the client device orientation. As another example, a size of the detected text may appear to become smaller as the text becomes further away, thereby indicating direction. As yet another example, an orientation of the text can help to determine whether it is being read from the front or the back.

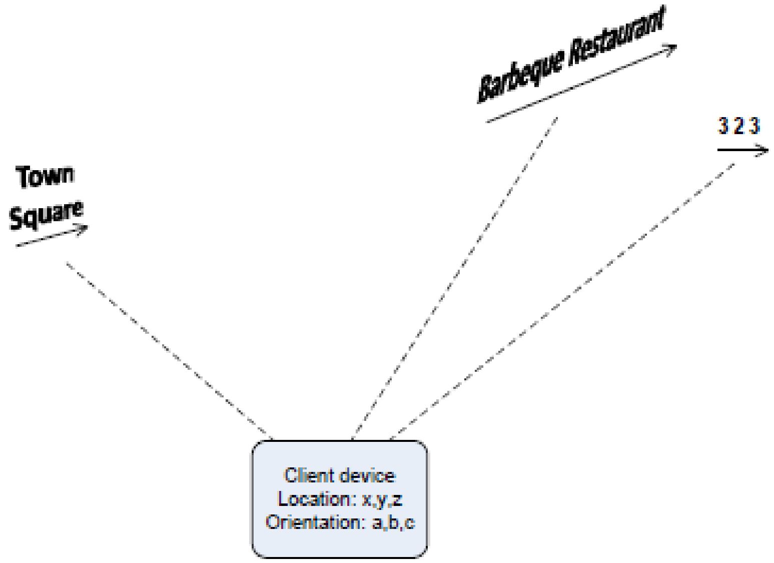

Fig. 2: Example of directional text information used to determine client device location and orientation.

Using the location information, including position and directional information as shown in the example of Fig. 2, a pose of the client device is determined. Such pose may include both location and orientation (yaw, pitch, roll). The pose may be determined using triangulation techniques.

Information regarding the image capture device, if known, may assist in the determination of pose information from the detected text. For example, if a focal length of the image capture device is known, assumptions may be made regarding the detected text. As an example of an assumption, information such as a distance between the image capture device and the detected text may be assumed. Using this information, less points of detected text are needed to determine a pose of the image capture device. By way of example only, whereas five pieces of text may be needed to determine the pose without knowing any information regarding the image capture device, only three or four pieces of text may be needed if the focal length of the image capture device is known.

In some examples, the client device may be trained to recognize logos, and use the logos to also determine pose. For example, the image capture device of the client device may detect a symbol representing a particular store, brand, etc. The symbol may be included in the 3D text database along with an associated location. Accordingly, the logo may be used in addition or in the alternative to text. In instances where only one or two pieces of text are visible, but several logos are more clearly visible, this may help to compute a more accurate pose.

The pose information may be determined to such a degree of accuracy and precision that such information can be used in other applications, such as augmented reality applications. In such applications, an optimal user experience is provided with a high degree of precision. For example, if the application is placing a virtual arrow pointing to a doorway, or a virtual label on a building, such virtual features will only be correctly placed if the client device pose is determined with a high level of accuracy.

In some examples, only partial matches of text may be used to identify pose information as described above. For example, referring to Fig. 3, an image is captured where some portion of a first piece of text is not visible. Specifically, as shown, whereas the first piece of text actually reads “1113,” in the image only “111” and a slight portion of the “3” is visible. Regardless of only partial visibility, the detected “111” partial text may be matched against the 3D text database.

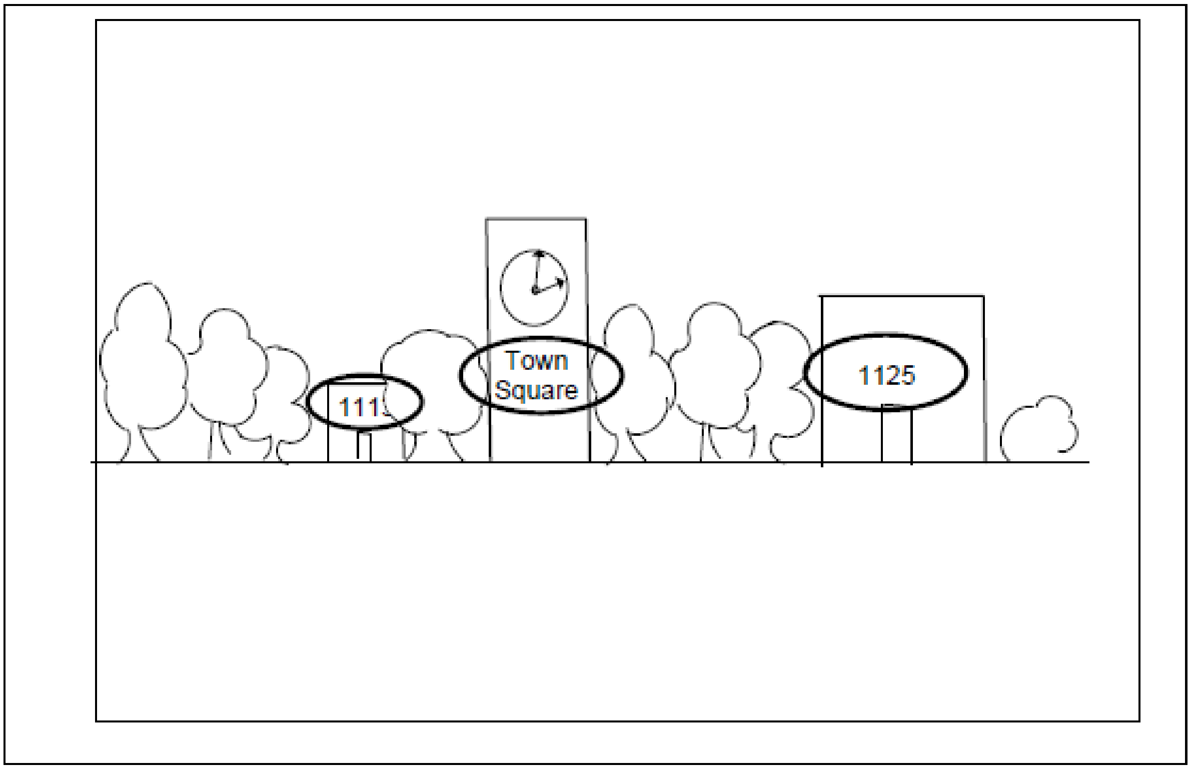

Fig. 3: Example captured image including partial text due to occlusion.

The partial match may result in multiple potential matches, particularly if a search range within the database is unrestricted. According to some examples, a search range within the database may be limited, such as by using known information. For example, if a general location of the client device is known, such as a particular city, neighborhood, or street, etc., then only matches within the database having corresponding locations may be searched. From the multiple potential matches, the correct match may be determined using further information, such as relational position of other text in the image, other objects in the image, etc.

Fig. 4 below illustrates an example of using the partial text, along with other detected text, to determine the location and orientation of the client device. Similar to the example of Fig. 2, each piece of detected text has associated location information, including position and direction information, in the 3D text database. Such location information is used to determine the client device pose, such as by using triangulation techniques.

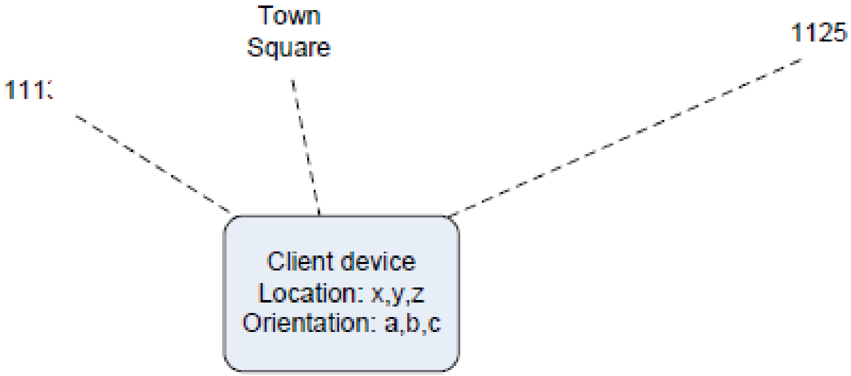

Fig. 4: Example of using partial text and other text to determine device location and orientation.

While in Fig. 3 the partial text was only partially visible due to occlusion, text may be only partially visible for other reasons. For example, only a portion of text may be included in the camera field of view. In examples where multiple camera images are captured, this partial text may be pieced together with the remaining text. For example, if the image capture device of the client device pans from left to right across a geographic area, text that was initially not visible or only partially visible, because it was past a right boundary of the image capture device field of view, may later become visible as the image capture device pans to the right and thus the boundaries change. In such an example, text detected at any point in the pan may be used, despite that it may no longer be visible by the end of the pan.

As mentioned above, a search area within the 3D text database may be limited using known information. In some examples, limiting the search region may also help to increase processing time. For example, conservative assumptions can be made about what is visible and what is not. As such, objects that are known to be farther away, or otherwise assumed to be not visible, may be excluded from the search region. In other examples, the 3D text database might also store a text size parameter and use that to reduce the search space. For example, large text may be included in the search region, even if far away, if it is still likely to be visible. Small text, on the other hand, may be excluded from the search region as it is less likely to be visible. Also the normal of the plane the text is on can be used to avoid text that is facing away from the client device, such as text on a back side of a billboard.

In some instances, text detected within a captured image may produce a false match. For example, if a delivery truck having text or a logo thereon is detected, such text may produce a match within the 3D text database. However, the associated location information for the text matching the truck is likely to be incorrect, because text that has moved during data collection is filtered out during building of the database. In such instances where false matches are produced, those false matches may be ignored. In some examples, the system may be trained to ignore text in captured images if that text is on a particular object, such as a truck, a T-shirt, or the like.

In some examples the client device may download a special map or localization layer for a city. In such examples, the map may indicate a raw location for the client device, which helps to narrow a range of possible detected poses for the device. In such examples, the image capture device of the mobile device is utilized as an extra sensor. Further sensors may also be utilized, such as light sensors, motion sensors, accelerometers, pedometers, gyroscopes, etc.

Alternative implementations of the text detection and matching described above are also possible, where more general features in the image are compressed and matched. Such objects may include, for example, buildings, landmarks, street signs, etc.

Using text, as opposed to objects or other information, to determine the pose of the image capture device is beneficial because the text is highly compressible. Accordingly, an entire database of text and associated locations may be downloaded to a client device without consuming too much memory. Fig. 5 illustrates an example where the 3D text database is downloaded onto the client device.

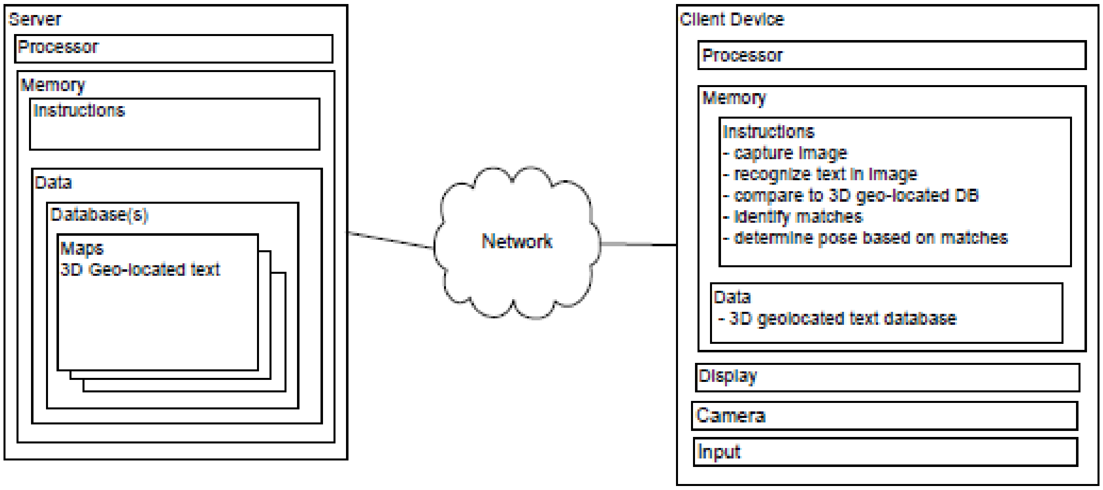

Fig. 5: Block diagram example of devices executing matching.

The 3D text database, once downloaded, may be used by the client device when a wireless communication signal is poor or non-existent. Thus, for example, where the network shown in Fig. 5 is unavailable, such as due to a poor connection, the client device may still seamlessly utilize applications, such as augmented reality applications, because its pose can be determined locally based on text in captured images.

The foregoing described application of using text to determine precise pose will also work indoors if there is enough text to perform a match. For example, it may be used to detect text in airports, malls, transit hubs, or other public indoor places. For privacy, however, it may not work in private indoor spaces, such as office buildings, residences, or other spaces in which data is not collected to form a 3D text database.

Using text for pose determination as described above is further advantageous, because the OCR recognition may be used to recognize text in any of a variety of languages. The text does not need to be understood, but only recognized so that it may be compared with the database. This may be particularly useful in cities where particular neighborhoods are devoted to a particular culture, such as New York City which includes Chinatown, Little Italy, etc. Even further, because text is typically placed so that it is highly visible, a significant number of datapoints is typically available to a client device, particularly in cities.