2-2015

# Scene Projection by Non-Linear Transforms to a Geo-Referenced Map for Situational Awareness

Kevin C. Krucki
*University of Dayton*

Vijayan K. Asari
*University of Dayton*, vasari1@udayton.edu

# Scene projection by non-linear transforms to a geo-referenced map for situational awareness

Kevin C. Krucki and Vijayan K. Asari

University of Dayton Vision Lab, Dayton, OH 45469 USA

## ABSTRACT

There are many transportation and surveillance cameras currently in use in major cities that are close to the ground and show scenes from a perspective point of view. It can be difficult to follow an object of interest across multiple cameras if many of these cameras are in the same area due to the different orientations of these cameras. This is especially true when compared to wide area aerial surveillance (WAAS). To correct this problem, this research provides a method to non-linearly transform current camera perspective views into real world coordinates that can be placed on a map. Using a perspective transformation, perspective views are transformed into approximate WAAS views and placed on a map. All images are then on the same plane, allowing a user to follow an object of interest across several camera views on a map. While these transformed images will not fit every feature of the map as WAAS images would, the most important aspects of a scene (i.e. roads, cars, people, sidewalks etc.) are accurate enough to give the user situational awareness. Our algorithm is proven to be successful when tested on cameras from the downtown area of Dayton, Ohio.

**Keywords:** Field of view, situational awareness, perspective transform, scene projection, horizon estimation

## 1. INTRODUCTION

There are many transportation and surveillance cameras currently in use in major cities across the world. Most of these cameras are close to the ground and show scenes from a perspective point of view. It can be difficult to follow an object of interest(e.g. a car or person) across multiple cameras if many of these cameras are in the same area due to the different orientations of these cameras. This is especially true when compared to wide area aerial surveillance (WAAS) views. To correct this problem, this paper provides a method to non-linearly transform current camera perspective views into real world coordinates that can be placed on a map in real-time. Using a perspective transformation matrix, original camera views are transformed into approximate WAAS views, called the approximate wide area view image (AWAVI) and placed into the cameras real world field of view (RWFOV) on a map. All images are then on the same plane, allowing a user to follow an object of interest across several cameras. While these transformed images will not fit every feature of the real world as WAAS images would, the most important aspects of a scene (i.e. roads, cars, people, sidewalks etc.) are accurate enough to give the user situational awareness.

The image transform takes place in multiple steps and on two markedly different types of images. The first type of image does not look above the horizon; the camera is looking only at the ground. The second type of image is from a camera looking above the horizon and, as can be seen in the Results section of the paper, is much more challenging to accurately fit a map compared to the first type of image. For the algorithm to determine which type of image we are looking at, certain information from the camera must be known. Knowledge of the tilt parameter allows us to determine if the camera is looking above the horizon at the sky. Obviously, the parts of the image that are sky cannot be represented as the ground on a 2D map and therefore are removed before transforming. To remove the sky from an image, horizon estimation, as used by Hoeim et. al.[1] is used to identify where the horizon is. All pixels above this location are discarded. This horizon estimation takes place by finding the intersections among all nearly parallel lines, also known as vanishing points, in the image and uses an average to find an approximate location that is accurate enough for the algorithm.

Further author information: (Send correspondence to K.C.K.)
K.C.K.: E-mail: kruckik1@udayton.edu
V.K.A: E-mail: vasari1@udayton.edu

Video Surveillance and Transportation Imaging Applications 2015, edited by Robert P. Loce, Eli Saber,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 9407, 94070Q · © 2015 SPIE-IS&T
CCC code: 0277-786X/15/$18 · doi: 10.1117/12.2077737

After the sky is removed from the second type of image, all images are treated equally. The image needs to then be transformed into its real world coordinates as if looking down on it in a map. The matrix needed for the transform to real world coordinates is attained by using the camera's RWFOV, which can be calculated by knowing the height, pan, tilt, zoom, and GPS coordinates of the camera and basic trigonometry. The camera's field of view in angular form, entirely different from its RWFOV, is the first parameter calculated from the equation. By knowing this angle, the tilt of the camera, and the camera's height off of the ground, we can calculate the length, base width and top width of the RWFOV using trigonometric relationships. Once the physical lengths and widths of the RWFOV are known, the distance the RWFOV is from the camera can be calculated by shifting it according to the tilt parameter of the camera. The four corners of the RWFOV are then calculated and placed around the camera's GPS location. The transformation matrix is calculated from these four corners and is applied to the original image to give the transformed, AWAVI image. The final result is obtained by placing the AWAVI onto the map.

## 2. METHODOLOGY

### 2.1 Real World Field of View

An example real world field of view (RWFOV) in 3-dimensional space can be seen in Fig. 1 in red. To calculate the four corners of the RWFOV, several different parameters from the camera must be known. This includes the camera's height above the ground, $h$, the maximum and minimum focal lengths of the camera, $f_{max}$ and $f_{min}$,respectively, the sensor width and sensor height of the camera, $d_x and d_y$, respectively, the GPS location of the camera in latitude and longitude, $G_X$ and $G_Y$, respectively, and the pan, tilt, and zoom parameters of the camera, $\beta$, $\gamma$, and $z$, respectively. It is important to note that pan and tilt are related to one another. When the pan is adjusted by 180˚and tilt is changed to be the same angle from straight down but on the other side, then the views are equal. For example, if straight down is considered $\gamma = -90$˚, then a camera that has $\gamma = -45$˚and $\beta = 0$˚is equal to a camera that has $\gamma = -135$˚and $\beta = 180$˚. It is also important to note that for this purpose, $\gamma = -90$˚when the camera is pointing straight down and $\beta = 0$˚when the camera is facing due north. With these parameters known, the RWFOV can be calculated using trigonometry.

First, the zoom parameter of the camera, $z$, must be taken into account. Changing $z$ will change the focal length of the camera, $f$ according to Eq. 1.

$$f = \frac{(f_{max} - f_{min})(z-1)}{z_{max}} + f_{min} \tag{1}$$

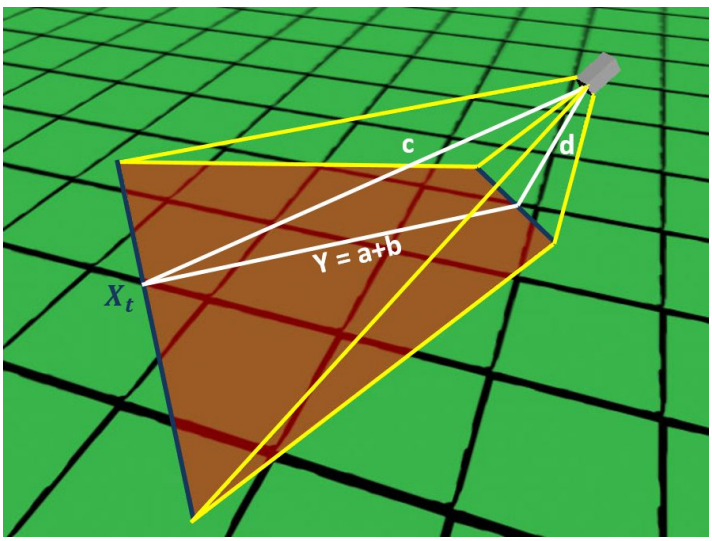



Figure 1: An example of the RWFOV is shown in red. The white lines show the triangle used to calculate the RWFOV (shown in Fig. 2), while the yellow lines represent the viewing limits of the camera in 3D space.

where $z_{max}$ is the maximum zoom setting of the camera.

With the focal length calculated, the angular fields of view of the camera $\alpha_x$ and $\alpha_y$ are calculated using Eqs. 2 and 3.[2]

$$\alpha_x = 2 * tan^{-1}(\frac{d_x}{2f}) \tag{2}$$

$$\alpha_y = 2 * tan^{-1}(\frac{d_y}{2f}) \tag{3}$$

Next, the distances $a$, $b$, $c$, and $d$ are calculated according to Fig. 2 and Eqs. 4-7. Fig. 2 is the same as the white portion of Fig. 1.

$$\theta = \alpha_y + \pi/2 - \gamma \tag{4}$$

$$\phi = \alpha_y - \theta \tag{5}$$

$$a = h * tan(\theta) \tag{6}$$

$$b = h * tan(\phi) \tag{7}$$

The distance the RWFOV reaches away from the camera, $Y$, corresponding to Fig. 1, is then calculated by Eq. 8.

$$Y = a + b \tag{8}$$

To calculate the base width $X_b$ and top width $X_t$, shown in blue in Fig. 1, the values for $c$ and $d$ are needed. These are calculated with Eqs 9 and 10 and $X_b$ and $X_t$ are calculated using Eqs. 11 and 12

$$c = h/cos(\theta) \tag{9}$$

$$d = h/cos(\phi) \tag{10}$$

$$X_b = 2d * tan(\alpha_x/2) \tag{11}$$

$$X_t = 2c * tan(\alpha_x/2) \tag{12}$$

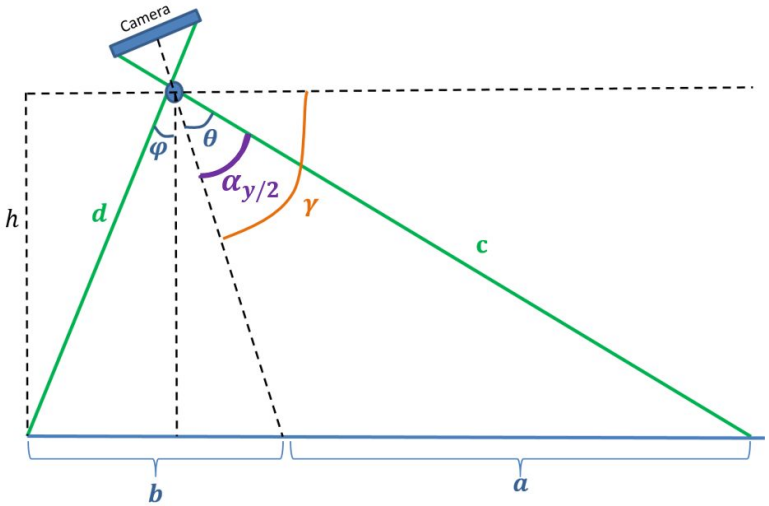



Figure 2: With knowledge of the camera parameters, including height, $h$, and tilt $\gamma$, and calculated parameters $\alpha_x$, $\theta$, and $\phi$, the distances $a$, $b$, $c$, and $d$, are calculated through basic trigonometry.

It is important to note that in many cases the camera points above the horizon, meaning that the $Y$ distance reaches out to infinity. In these cases, the distances $X_t$ and $Y$ are manually scaled back to a maximum value of the user's preference.

The distances of the RWFOV are now known. However, its location with respect to the camera is unknown. Therefore, a tilt factor, $t_f$, to move the RWFOV away from the camera is needed. Eq. 13 is used to calculate $t_f$.

$$t_f = 2c * tan(\alpha_x/2) \tag{13}$$

All distances are now calculated to find the four corners of the RWFOV in relation to point (0,0) i.e. the camera is at point (0,0). The points $P_i(x_i, y_i)$, where $i = 1, 2, 3, 4$, are the four points of the RWFOV if the camera is centered at (0,0), and $x_i$ and $y_i$ are the locations of the points in 2D space. These are detailed in Eqs. 14-17.

$$P_1(X_b/2, Y/2 + t_f) \tag{14}$$

$$P_2(-X_b/2, Y/2 + t_f) \tag{15}$$

$$P_3(X_t/2, -Y/2 + t_f) \tag{16}$$

$$P_4(X_t/2, -Y/2 + t_f) \tag{17}$$

These points are then rotated according to the pan parameter, $\gamma$, around the camera. This is accomplished by transforming the points into polar coordinates, rotating each point by $\gamma$ and then transforming back into Cartesian coordinates. Lastly, these points need to be converted from the unit that height is in (e.g. feet, meters, miles) into degrees of longitude and latitude and placed at the camera's GPS location. An overview of this process can be seen in Fig. 3 and an example of the final RWFOV on a map with parameters listed in the caption can be seen in Fig. 4.

## 2.2 Transforming the Image

To get the approximate wide area view image (AWAVI), several operations are performed on the original image. To start, we determine what type of image is being transformed: an image with sky or an image without sky. Knowledge of the tilt parameter and angular field of view (FOV) of the camera allows us to determine which type of image we are transforming. If the angular FOV is below the horizon then there will not be sky in the image and we can immediately start the transform. If the angular FOV is above the horizon then there will be sky in the image. The part of the image that is sky cannot be represented as the ground on a 2D map and therefore it is removed from the image using horizon estimation.

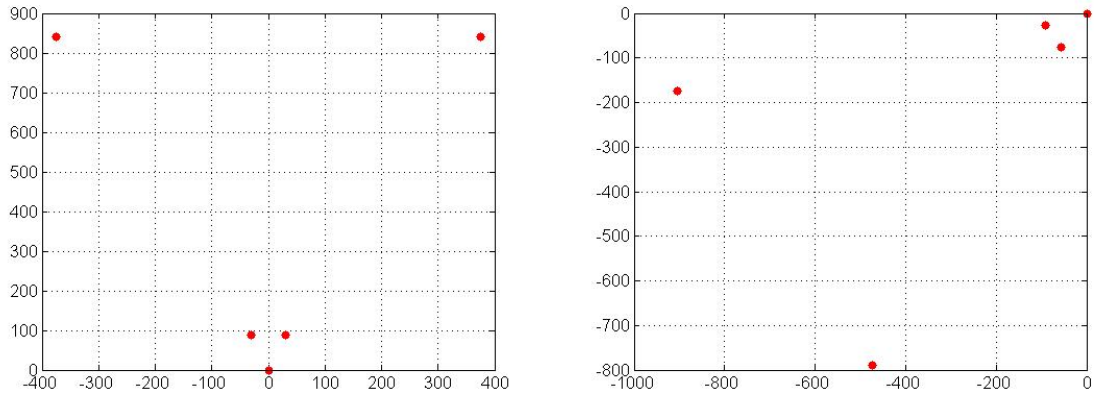
### 2.2.1 Horizon Estimation

The horizon can be estimated in images using the features of the image. It can be accomplished by finding the intersections of long nearly parallel lines in the image. These lines are found using edge detection in the same manner Kosecka and Zhang[3] did. Then the position that minimizes the $\frac{1}{2}L$ distance from these intersections is calculated. This provides for a strong estimate of where the horizon exists in man-made scenes because they contain many lines parallel to the ground plane. These lines intersect at vanishing points which exist on the horizon.[1] An example of this can be seen in Fig. 5.

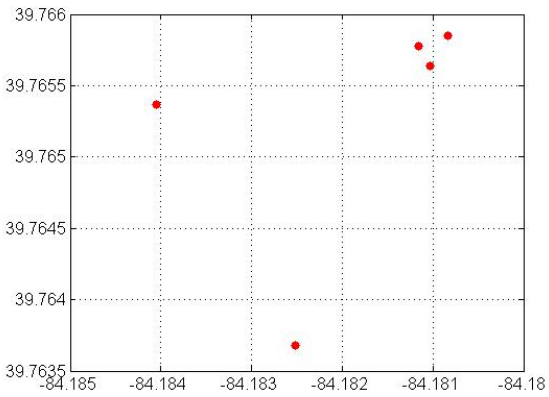
### 2.2.2 Transforming the Image into the RWFOV

Now that the sky has been removed from the image, the transform process can begin. In order to get a top down view of the image it must be transformed into the RWFOV. To do this, the four corners of the image $(x_i, y_i)$ must be plotted to the four corners of the RWFOV $(u_i, v_i)$. This is possible by calculating the transformation matrix, $H$ shown in Eq. 18, and solving the system of linear equations given by Eq. 19.[4]

$$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \tag{18}$$




(a) (b)




(c)

Figure 3: An overview of the process of calculating the RWFOV. Fig. 3a shows a plot of an example RWFOV with the camera at (0,0), and no rotation, translation, or scaling. Fig. 3b shows a plot of an example RWFOV with the camera at (0,0), rotation, and no translation or scaling. Fig. 3c shows a plot of an example final RWFOV with the camera at its GPS coordinates. Note that the shape in Fig. 3c is different than previously due to the different scaling of degrees of longitude and latitude.

$$H \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} \tag{19}$$

If it is assumed $h_9 = 1$, then Eq. 19 will give eight equations for eight variables and $H$ is solvable. Using MATLAB's imtransform with $H$ and the image will then produce the AWAVI image.

## 3. RESULTS

The dataset used to test this algorithm was collected from cameras in downtown Dayton, Ohio controlled by The Institute for Development and Commercialization of Advanced Sensor Technology (IDCAST). After analysis of different images from IDCAST cameras, it became apparent that this approach was not perfect for all images. If the camera had a large RWFOV or was aimed above the horizon, a large amount of distortion occurred. Therefore, the base of the camera field of view was widened in the transform to remedy this distortion. The camera used for all results had the parameters $f_{max} = 119$, $f_{min} = 3.3$, and $z_{max} = 72$, $d_x = 3.2$ and $d_y = 2.4$.

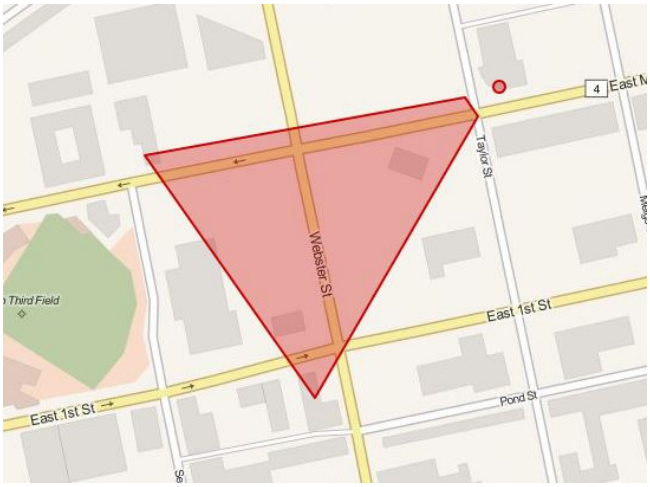



Figure 4: Example RWFOV placed onto a map. The parameters for this example are $f_{max} = 119$, $f_{min} = 3.3$, $z_{max} = 72$, $d_x = 3.2$, $d_y = 2.4$, $z = 1.923$, $\gamma = -9$, $\beta = -125\circ$, $G_X = -84.18084$, $G_Y = 39.7658$, $h = 35$ feet and a maximum length of $y$ and $X_t$ of 752 feet.

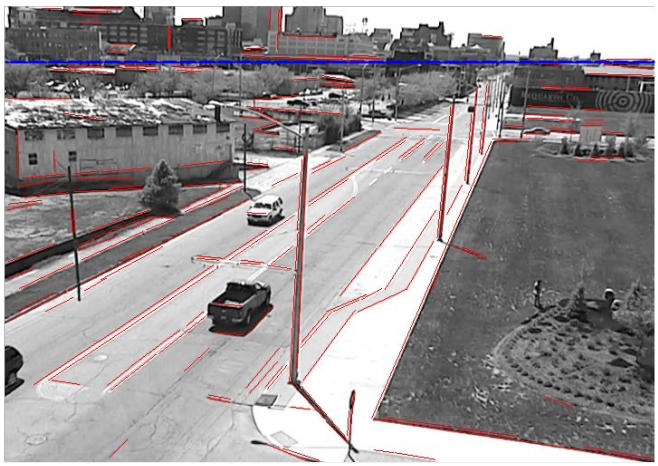


Figure 5: Example of horizon estimation to remove the sky. The red lines show nearly parallel lines which intersect at the vanishing points of the scene. The blue line shows the estimated horizon.

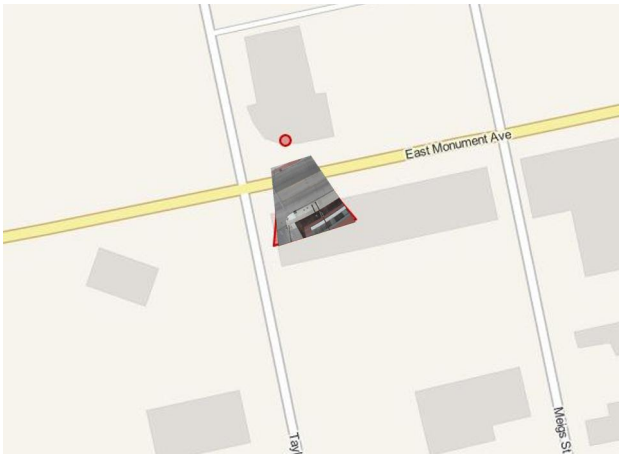
Several results are shown Figs. 6-10 with the following images: the original image, the RWFOV, the transformed image, and the transformed image placed on a map. The camera parameters are in the caption of each figure. Results for images without sky are shown in Figs. 6, 7, and 8. Results for these images are quite good, as most of the image looks as it would if we were looking straight down from a plane. The only blemishes in the images are in Figs. 6 and 8 the building appears to be ground and in Fig. 7 the street lights stretch across the image. Otherwise, the features of the scene, especially the road and cars, are transformed nicely. These images give a phenomenal amount of situational awareness, as there is now a directional component to what is happenning in the image. Even though they do not precisely match the map, we can still see what is happenning in the scene.

On the other hand, the results shown for images with sky in Figs. 9 and 10 have large amounts of distortion near the top of their transformed images. Although the parts of the image that are close to the camera look acceptable, the area furthest from the camera looks too distorted to get any reliable information. This is due to the large size of the RWFOV stretching out pixels at the top of the image. To fix this issue, the base of the RWFOV was tripled. The effect of tripling the base width is shown in Figs. 11 and 12 with their original images as well. Although these images do not line up with the road as well as the original transformation images do,

(a) Original Image


(b) Real World Field of View of Camera

(c) Transformed Image




(d) Transformed Image Placed on a Map

Figure 6: The camera parameters for this example are $z = 1.923$, $\gamma = -36.5375$, $\beta = 162.0796\circ$, $G_X = -84.18084$, $G_Y = 39.7658$, $h = 35$ feet and a maximum length of $y$ and $X_t$ of 752 feet. The original image, RWFOV of the camera, transformed image, and transformed image placed on the map are shown.

they are much clearer at the furthest point from the camera. The cars in Fig. 11 are an excellent example of this.

## 4. CONCLUSION

This paper presented a novel way to look at regular images from a camera on a map. First the RWFOV of a camera was calculated with knowledge of camera parameters and trigonometry. Next, the transformation matrix was calculated by solving a system of linear equations and the image was transformed. Lastly, the image was placed onto a map to give situational awareness. This was proven in the results section, where several different images collected in downtown Dayton, Ohio were transformed into their RWFOV. While all transformed images gave a superior amount of situational awareness compared to the original image, some more closely resembled

the real world than others. The images that were transformed into small RWFOVs had much less distortion than images with large RWFOVs or views out to infinity. To combat the distortion that occurs in an image transformed into a large RWFOV the base of the RWFOV was arbitrarily tripled. Although these images did not fit the real world as closely, they were much clearer than the previous transformations which led to more situational awareness. All image transformations effectively gave situational awareness and, if live video was being streamed into the RWFOV, it would be much simpler for a person to follow a car or person between different camera views.

## REFERENCES


[1] D. Hoiem, A. E. and Hebert, M., "Automatic photo pop-up," in [*ACM SIGGRAPH*], **Volume 24 Issue 3**, Pages 577–584 (July 2005).

[2] McCollough, E., "Photographic topography," *Industry: A Monthly Magazine Devoted to Science, Engineering and Mechanic Arts* (1893).

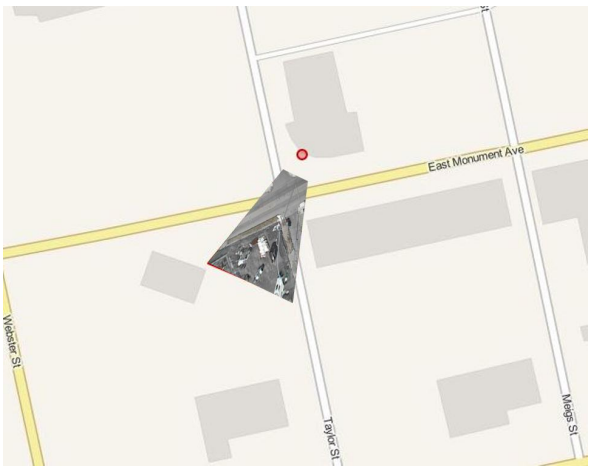
[3] Kosecka, J. and Zhang, W., "Video compass," in [*ECCV*], Springer-Verlag (2002).

[4] Hartley, R. I. and Zisserman, A., [*Multiple View Geometry in Computer Vision*], Cambridge University Press, 2nd ed. (2004).

(a) Original Image


(b) Real World Field of View of Camera

(c) Transformed Image

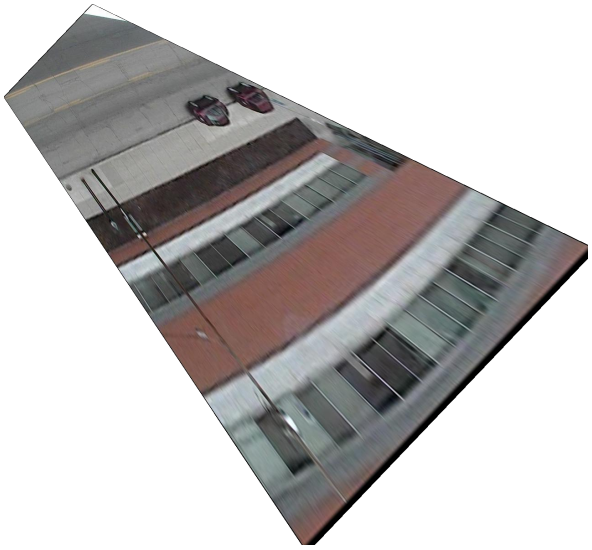



(d) Transformed Image Placed on a Map

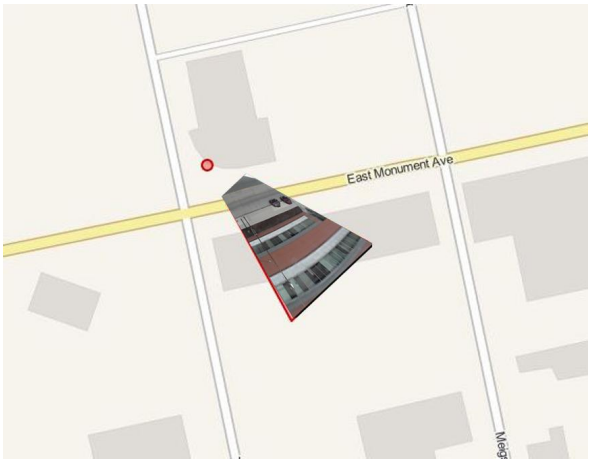
Figure 7: The camera parameters for this example are $z = 1.923$, $\gamma = -156.9143$, $\beta = -156.9143\circ$, $G_X = -84.18084$, $G_Y = 39.7658$, $h = 35$ feet and a maximum length of $y$ and $X_t$ of 752 feet. The original image, RWFOV of the camera, transformed image, and transformed image placed on the map are shown.

(a) Original Image


(b) Real World Field of View of Camera



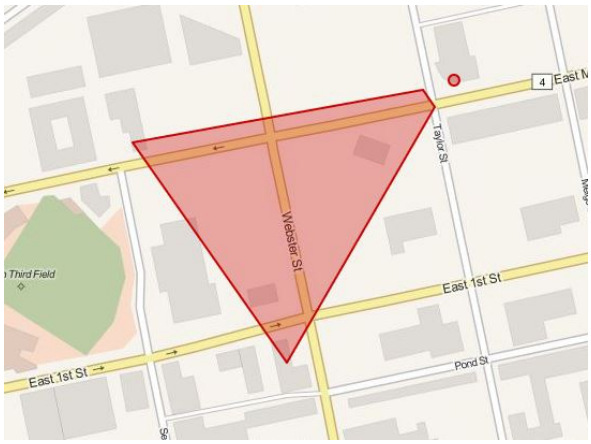
(c) Transformed Image



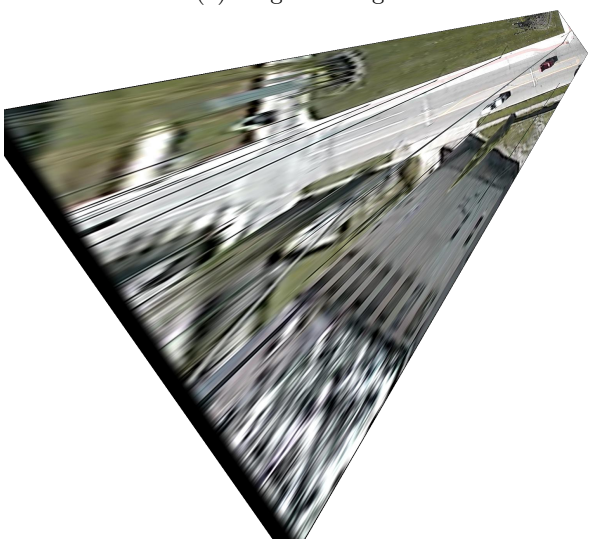

(d) Transformed Image Placed on a Map

Figure 8: The camera parameters for this example are $z = 1.923$, $\gamma = -22.7709$, $\beta = 133.6478\circ$, $G_X = -84.18084$, $G_Y = 39.7658$, $h = 35$ feet and a maximum length of $y$ and $X_t$ of 752 feet. The original image, RWFOV of the camera, transformed image, and transformed image placed on the map are shown.

(a) Original Image




(b) Real World Field of View of Camera



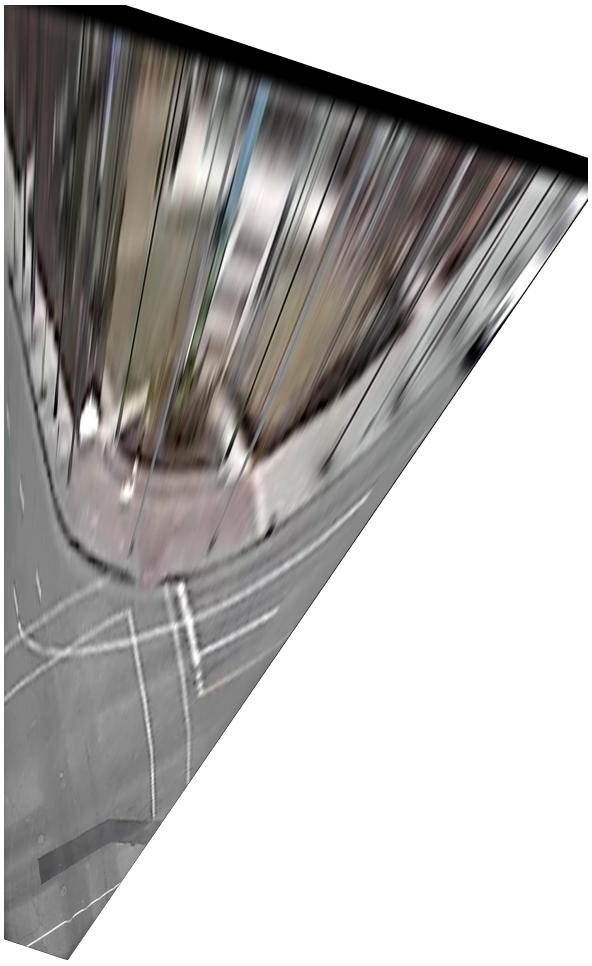
(c) Transformed Image


(d) Transformed Image Placed on a Map

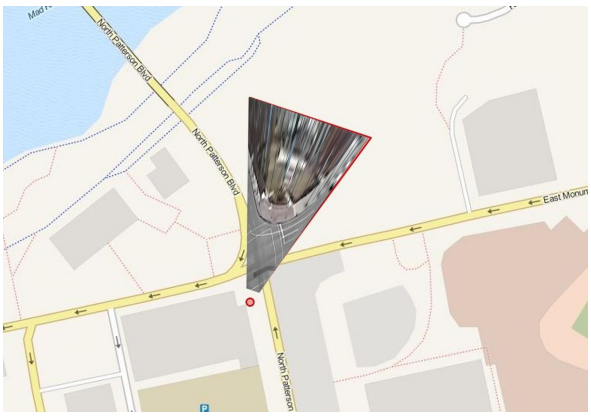
Figure 9: The camera parameters for this example are $z = 1.923$, $\gamma = -9$, $\beta = -125\circ$, $G_X = -84.18084$, $G_Y = 39.7658$, $h = 35$ feet and a maximum length of $y$ and $X_t$ of 752 feet. The original image, RWFOV of the camera, transformed image, and transformed image placed on the map are shown.

(a) Original Image


(b) Real World Field of View of Camera



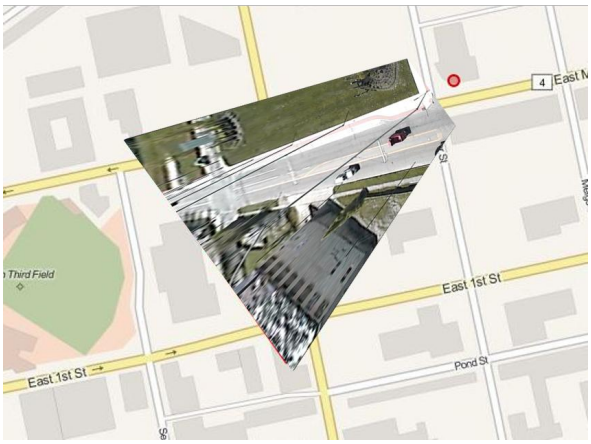
(c) Transformed Image




(d) Transformed Image Placed on a Map

Figure 10: The camera parameters for this example are $z = 1.923$, $\gamma = -16.7996$, $\beta = 18.2731\circ$, $G_X = -84.187813$, $G_Y = 39.764368$, $h = 18$ feet and a maximum length of $y$ and $X_t$ of 752 feet. The original image, RWFOV of the camera, transformed image, and transformed image placed on the map are shown.

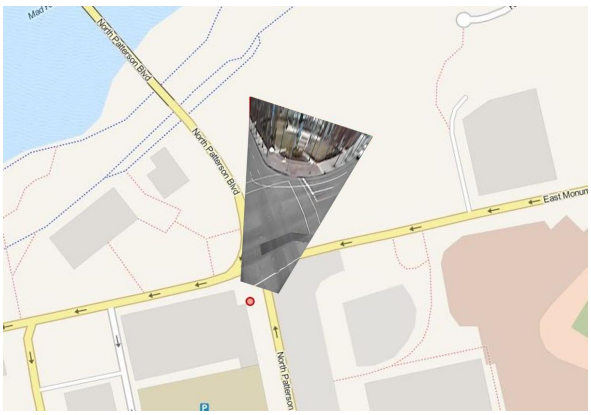
(a) Original Image




(b) Real World Field of View of Camera

Figure 11: The original image and its transformed image with the base tripled.

(a) Original Image




(b) Real World Field of View of Camera

Figure 12: The original image and its transformed image with the base tripled.