Inverse Perspective Transformation for Video Surveillance

Theo E. Schouten[a], and Egon L. van den Broek[b]

[a]Institute for Computing and Information Science (ICIS), Radboud University Nijmegen
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands
T.Schouten@cs.ru.nl http://www.cs.ru.nl/T.Schouten

[b]Center for Telematics and Information Technology (CTIT), University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands
vandenbroek@acm.org http://eidetic.ai.ru.nl/egon

ABSTRACT

In this research, we are considering the use of the inverse perspective transformation in video surveillance applications that observe (and possible influence) scenes consisting of moving and stationary objects; e.g., people on a parking area. In previous research, objects were detected on video streams and identified as moving or stationary. Subsequently, distance maps were generated by the Fast Exact Euclidean Distance (FEED) transformation, which uses frame-to-frame information to generate distance maps for video frames in a fast manner. From the resulting distance maps, different kinds of surveillance parameters can be derived. The camera was placed above the scene, and hence, no inverse perspective transformation was needed. In this work,the case is considered the case that the camera is placed under an arbitrary angle on the side of the scene, which might be a more feasible placement than on the top. It will be shown that an image taken from a camera on the side can be easily and fast converted to an image as would be taken by a camera on the top. The allows the use of the previously developed methods after converting each frame of a video stream or only objects of interest detected on them.

Keywords: inverse perspective transformation, video surveillance, distance maps/transforms, Fast Exact Euclidean Distance (FEED)

1. INTRODUCTION

In video surveillance, it is often needed to identify objects, to detect their possible movements and to observe that their relative positions an distances stay within the set limits. Unwanted or possible dangerous distances should be automatically detected (e.g., too small or too large), in order to take appropriate actions. The automation of this process is of the utmost importance since the vigilance of humans is limited. The ability of humans to hold attention and to react to rarely occurring events, is extremely demanding and, consequently, prone to error due to lapses in human attention[6]. These actions might include signaling a (human) supervisor or sending relevant distance information to objects (or persons) that have capabilities for changing the course of their actions[1,3,6].

In our previous research an operational environment was chosen where a single camera provides a top view of a scene with moving and stationary objects. The calculation of distances was based on the Fast Exact Euclidean Distance (FEED) transformation developed by Schouten and Van den Broek[10]. This Euclidean Distance Transformation converts a binary image to another image of the same size, such that each pixel has a value equal to the *exact* Euclidean distance to the nearest object pixel, where most other algorithms approximate the true Euclidean distance[4,5]. The new image is called the *exact* Euclidean Distance Map (E^2DM) of the old image. With the calculation of the E^2DM for stationary objects, the distance of a moving object to the stationary object can be calculated by taking the minimum of the distances in this E^2DM at the locations of the border pixels of the moving object.

As shown by Schouten, Kuppens, and Van den Broek[8], the E^2DM of binary images containing stationary and moving objects can be obtained very fast, using the FEED transformation. Here, very fast denotes that the E^2DMs of a sequence of more than a few images can be obtained faster than city-block[7] or 3,4 chamfer[2] distance maps. To achieve the latter, the E^2DMs for the stationary objects ($E^2DM_{stationary}$) and for the moving object

Computational Imaging VI, edited by Charles A. Bouman, Eric L. Miller, Ilya Pollak,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6814, 681415, © 2008 SPIE-IS&T · 0277-786X/08/$18

(E^2DM_{moving}) are calculated separately and combined, using the minimum operator, to obtain the distance map for the total image:

$$E^2DM_{stationary+moving}(p) \;=\; \min\{E^2DM_{stationary}(p)\ ,\ E^2DM_{moving}(p)\} \tag{1}$$

This E^2DM is used to provide each object (e.g., a robot) with processing capabilities, with a map of objects and free space in its environment. The robot can use information obtained from the E^2DM to achieve its goal; e.g., path planning[11].

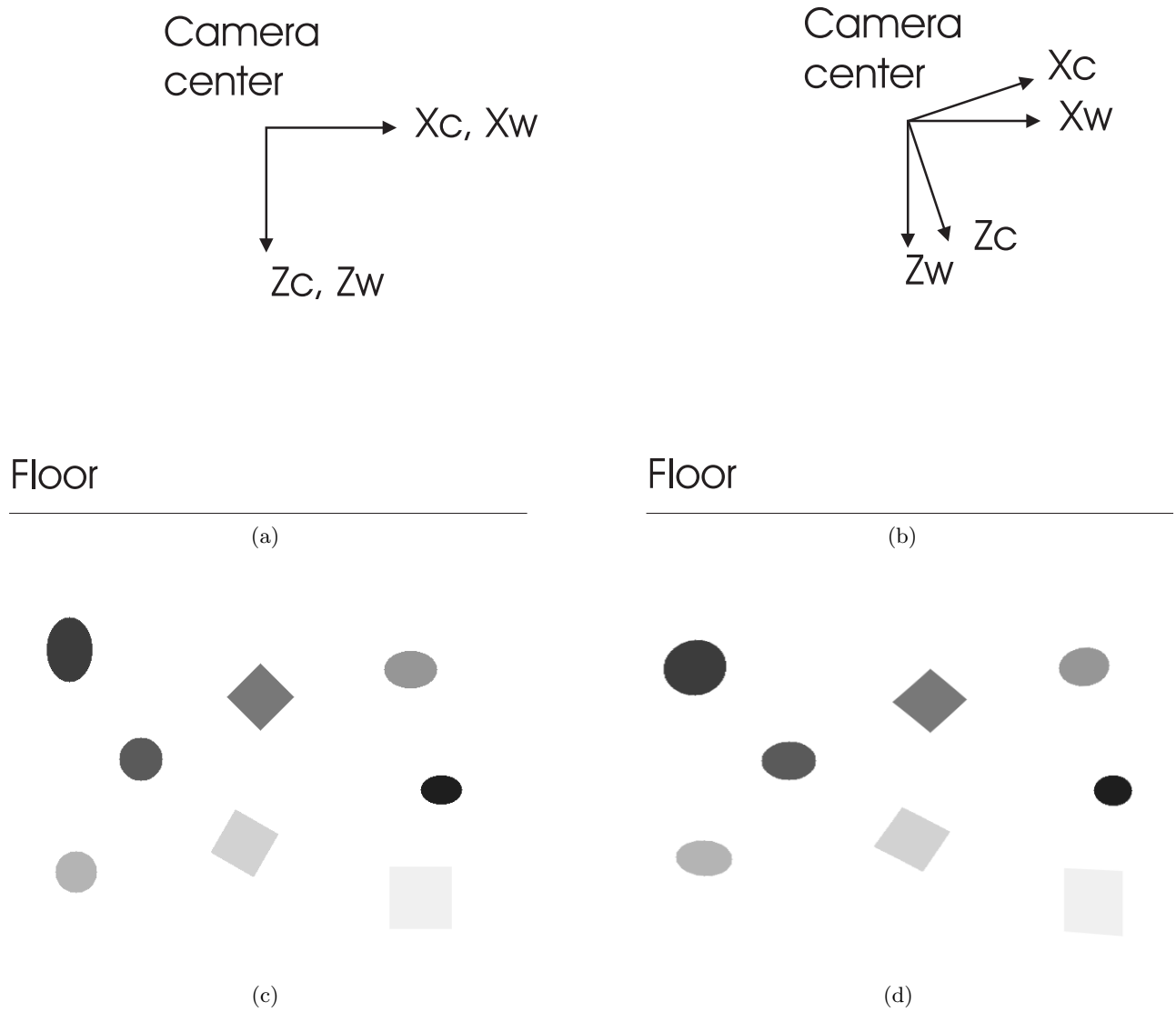


Figure 1: Camera placed on the top (a) or side (b) on a scene. In (c) a possible image taken from the top is shown where objects each have a different gray level. In (d) the same image is shown but now taken from the side.

Further Schouten, Kuppens, and Van den Broek[9] developed a system to detect and locate stationary and moving object in a video stream. The conversion of the gray and color images of the camera to binary (i.e., black-and-white) images, which are used for further processing, was not discussed. However, they take into account that this conversion process is not able to handle all the challenges posed by changing illumination conditions, shadows and video noise correctly. This conversion and its imperfections was modeled by a virtual, dynamic robot navigation environment.

For video surveillance application placement of a camera on top of a scene is often not practical, a position of the camera on the side looking down at the scene under an angle is then more feasible. Images taken by such a camera have a distorted geometry due to the perspective projection of the lens system of the camera. Therefore in this paper a description is given of a method to invert the perspective transformation such that an image taken from the side can be converted to an image taken from the top. It will be shown that this can be done with a simple transformation. The previously developed methods for video surveillance can than be applied.

In Section 2 the equations to invert the perspective transformation are derived. The implementation of the developed method and tests on generated images are described in Section 3. This is followed by tests on real images taken by a digital camera which are described in Section 4. In the last section some conclusions are given.

2. SIDE TO TOP VIEW CONVERSION

In Figure 1 the two possible camera positions, top and side, are visualized. The top position is shown in Figure 1(a) and a possible image taken in that set-up is given in Figure 1(c). The Xc and Zc axis, the Yc axis goes perpendicular into the drawing plane, define the orientation of the camera, with the usual convention that the image is taken in the Zc direction to provide an image from the scene on the floor. In contrast Figure 1(b) gives the side position of the camera and Figure 1(d) shows the same scene as in Figure 1(c) but now taken with the camera in the side position. Note that a perspective geometric distortion is present in Figure 1(b). Note also that only a rotation around the Y axis is taken into account. In this section a method is given to convert the image taken from the side into an image taken from the top.

Assume a pinhole lens with the image plane a distance f from the camera center and let s be the size of a rectangular pixel in the same unit as in which all the coordinates are given. Then the pixel coordinates (x_i, y_i) in the top camera position of a point (x_w, y_w, z_w) in the world coordinate system are given by:

$$\begin{array}{ll} (1) & x_i \;=\; (f/s)\; x_w \;/\; z_w \\ (2) & y_i \;=\; (f/s)\; y_w \;/\; z_w \end{array} \tag{2}$$

Let α be the angle over the camera is rotated around the Yc axis then the coordinates of the same point (x_w, y_w, z_w) in the camera coordinate system (x_c, y_c, z_c) are given by:

$$\begin{array}{ll} (1) & x_c \;=\; x_w\; cos\; \alpha \;-\; z_w\; sin\; \alpha \\ (2) & y_c \;=\; y_w \\ (3) & z_c \;=\; -\,x_w\; sin\; \alpha \;+\; z_w\; cos\; \alpha \end{array} \tag{3}$$

Then the pixel coordinates (u_i, v_i) in the side camera position of the point (x_w, y_w, z_w) are given by:

$$\begin{array}{ll} (1) & u_i \;=\; (f/s)\; x_c \;/\; z_c \\ (2) & v_i \;=\; (f/s)\; y_c \;/\; z_c \end{array} \tag{4}$$

Substituting the above equations into each other working toward expressing the side view coordinates (u_i, v_i) as function of the top view coordinates (x_i, y_i) of the same point in the world, the following equations are obtained:

$$\begin{array}{ll} (1) & u_i \;=\; (\; x_i \;-\; (f/s)\; tan\; \alpha\;) \;/\; (\; 1 \;+\; (s/f)\; x_i\; tan\; \alpha\;) \\ (2) & v_i \;=\; y_i \;/\; (\; (s/f)\; x_i\; sin\; \alpha \;+\; cos\; \alpha\;) \end{array} \tag{5}$$

In the above equation only pixel coordinates and the properties (s, f) and orientation (α) are present. The world coordinates completely drop out in the above equation, this makes the conversion from a side view image to a top view image feasible.

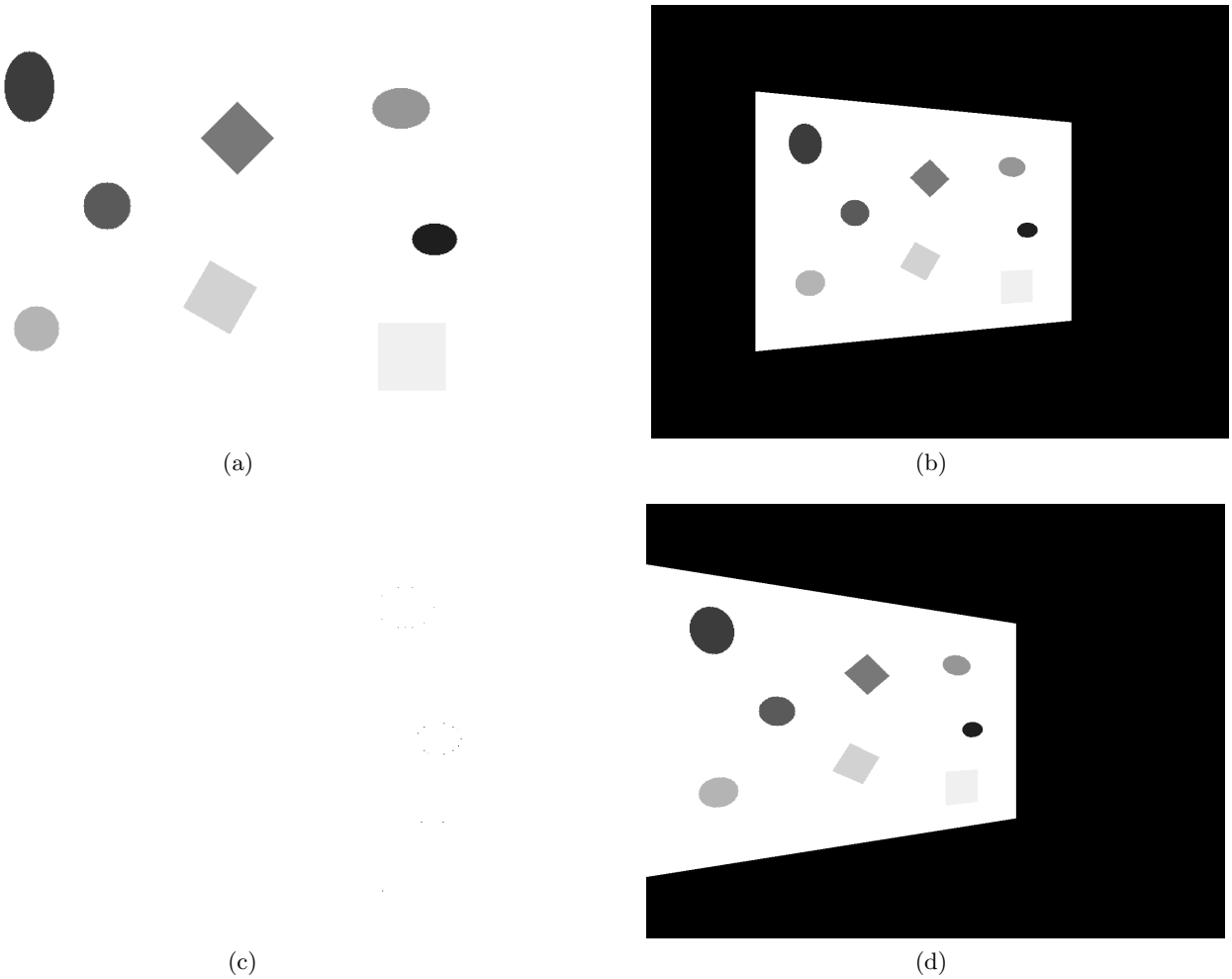
Remark that the image coordinates in the above equations are relative to the point where the Zc axis intersects the image plane. This has to be taken into account in the computer programs doing the conversion, in general the coordinates of the center of an image provide the center point unless the image is cropped.

The inverse transformation can also easily be derived and is given by:

$$\begin{array}{ll} (1) & x_i \;=\; (\; u_i \;+\; (f/s)\; tan\; \alpha\;) \;/\; (\; 1 \;-\; (s/f)\; u_i\; tan\; \alpha) \\ (2) & y_i \;=\; v_i\; (\; (s/f)\; x_i\; sin\; \alpha \;+\; cos\; \alpha\;) \end{array} \tag{6}$$

The above equation can be used for testing purposes to ensure that the conversions are implemented correctly. Further to test the artificial image effects introduced by the rounding or truncating when going from floating point pixel coordinates provided by Equations 5 and 6. Depending on the needed accuracy interpolating and/or averaging methods can then be introduced.

3. IMPLEMENTATION



(a) (b) (c) (d)

Figure 2: (a): Original generated image, the camera is placed on the top. (b): Image calculated using Equation 6, showing the image under an angle of 15°. (c): The difference between the image in (b) corrected to a top view and the original image in (a). Only a few points with a different intensity values are visible, shown as black points. (d): The image generated under an angle of 25°.

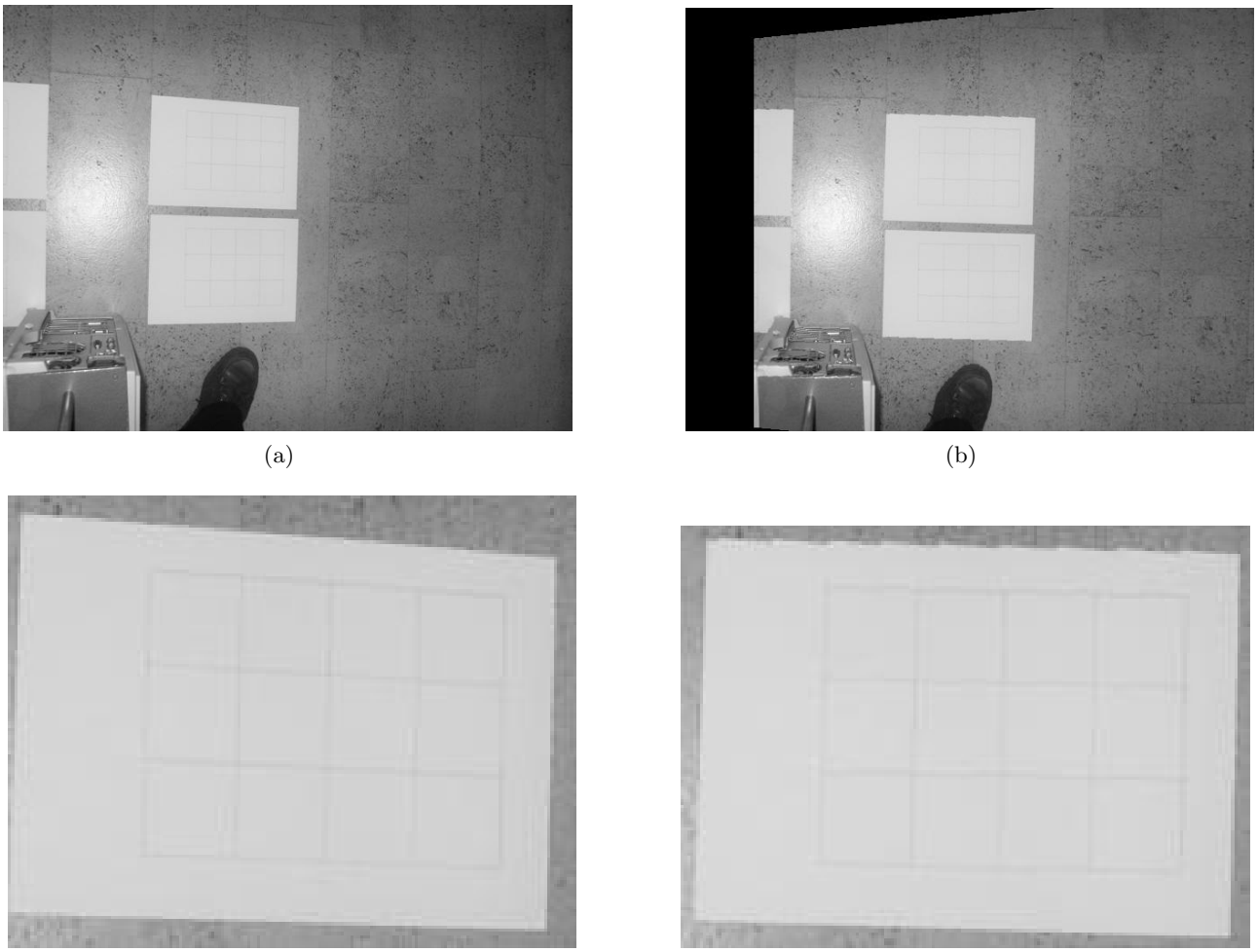
The conversion from the side to the top view was implemented in an usual and simple way. First the dimensions of the top view image are chosen depending on the application, e.g. large enough to use the whole area on the floor of the side view image or selecting a smaller size to cover only an area of interest. Then a raster scan is made over the top view image, computing for each integer pixel coordinate (x_i, y_i) using equation 5 the corresponding side view pixel coordinate (u_i, v_i) as floating point values. This takes only a small number of arithmetic operations per pixel. These values are then rounded to integer values to give the pixel in side view image from which the intensity or color value for the top view image is taken.

Also the inverse conversion from top to side view was implemented in the same way. Figure 2(a) shows a generated top view image from which a side view image under an angle of 15° was calculated, see Figure 2(b).

That image was than converted back to a top view image and the difference with the original image is shown in Figure 2(c). For only a few pixels the intensity values are different. Further Figure 2(d) shows a side view image under an angle of 25°, this shows that the distortion is larger. For the side view images sizes twice as large in each dimension were taken, pixels not having corresponding points in the image they are generated from, are set to black.

Various other generated images were tested in the above way, they all had very little pixels with a deviating intensity due to computational problems.

4. TEST ON IMAGES



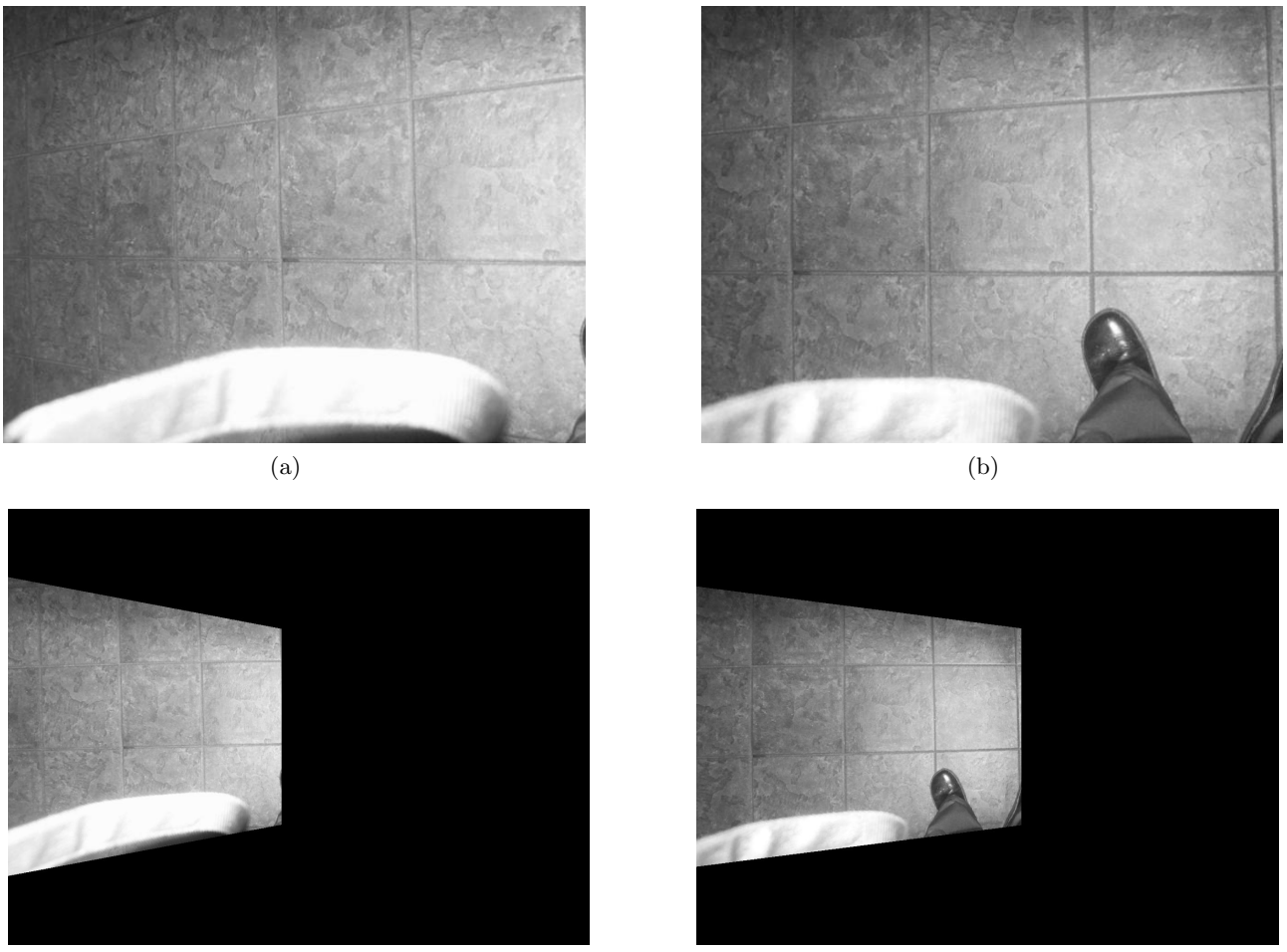
(a) (b)

(c) (d)

Figure 3: (a): Original image, taken with a digital camera under an angle of 15°. (b): Image converted to a top view using Equation 5. (c): Details of the top left A4 paper of the original image (d): The same details from the corrected top view image

Test images were taken with a digital camera placed in a simple mounting system. In Figure 3(a) one of the images is shown taken under an angle of 15° as determined with a simple ruler. The image plane distance

f (see Section 2) was known from the camera specification (7,15 mm) and the size s of a pixel was estimated from measurements on images from a grid (as shown in Figure 3 to be 0.0112 mm. Using this information the top view image calculated from Figure 3(a) is shown in Figure 3(b). Details from the top left A4 size paper are shown in the bottom images. Although a bit hard to judge, the squares on paper are nearly square on the top view image while not on the original image, and show maybe only a little more artifacts than on the original image. The latter appear to extent over not more than one or mostly two pixels. The same qualitatively results were observed on other images.



(a) (b)

(c) (d)

Figure 4: (a) and (b): Original images, taken at angles of around 30° resp. 20°. (c) and (d): Their calculated top view images.

In Figure 4 some more side view images are shown together with the calculated to view images, showing again that the perspective distortion is removed.

The obtained quality of the method is certainly good enough for most of the video surveillance, where the lens and other factors introduce larger errors than the inverse perspective method developed in this paper.

5. DISCUSSION

In this paper a simple method is developed to convert an image taken with a camera on the side of a scene to an image taken from the top of the scene. This methods corrects the geometric distortion caused by the perspective transformation. The accuracy was shown to be in the order of one to two pixel or less, which is good enough for

video surveillance purposes. The method takes only a small number (7) of arithmetic operations per pixel, which might be (partially) replaced by table lookup if that turns out to be faster on the execution platform used.

The method might be used on full images of frames of a video stream, allowing previous developed methods and systems for top views to be used unchanged. It might also be used after e.g. detection of relevant objects on distorted images to only correct relevant parts of images or video streams.

REFERENCES

1. A. Amer and C. Regazzoni. Introduction to the special issue on video object processing for surveillance applications. *Real-Time Imaging*, 11(3):167–171, 2005.
2. G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing: An International Journal*, 34:344–371, 1986.
3. A. R. Dick and M. J. Brooks. Issues in automated visual surveillance. In *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA 2003)*, pages 195–204, Sydney, Australia, December 2003.
4. C. Fouard and G. Malandain. 3-D chamfer distances and norms in anisotropic grids. *Image and Vision Computing*, 23(2):143–158, 2005.
5. A. Hajdu and L. Hajdu. Approximating the Euclidean distance using non-periodic neighbourhood sequences. *Discrete Mathematics*, 283(1–3):101–111, 2004.
6. A. Hampapur, L. M. Brown, J. Connell, M. Lu, H. Merkl, S. Pankanti, A. W. Senior, C.-F. Shu, and Y.-L. Tian. Multi-scale tracking for smart video surveillance. *IEEE Transactions on Signal Processing*, 22(2):38–51, 2005.
7. A. Rosenfeld and J. L. Pfaltz. Distance functions on digital pictures. *Pattern Recognition*, 1:33–61, 1968.
8. T. E. Schouten, H. C. Kuppens, and E. L. van den Broek. Timed Fast Exact Euclidean Distance (tFEED) maps. *Proceedings of SPIE (Real Time Imaging IX)*, 5671:52–63, 2005.
9. T. E. Schouten, H. C. Kuppens, and E. L. van den Broek. Three dimensional fast exact euclidean distance (3D-FEED) maps. *Proceedings of SPIE (Vision Geometry XIV)*, 6066:60660F, 2006.
10. T. E. Schouten and E. L. van den Broek. Fast Exact Euclidean Distance (FEED) Transformation. In J. Kittler, M. Petrou, and M. Nixon, editors, *Proceedings of the 17th IEEE International Conference on Pattern Recognition (ICPR 2004)*, volume 3, pages 594–597, Cambridge, United Kingdom, 2004.
11. F. Y. Shih and Y.-T. Wu. Three-dimensional Euclidean distance transformation and its application to shortest path planning. *Pattern Recognition*, 37(1):79–92, 2004.