UNIVERSITÄT AUGSBURG
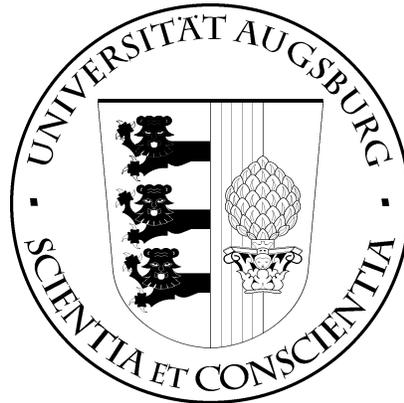

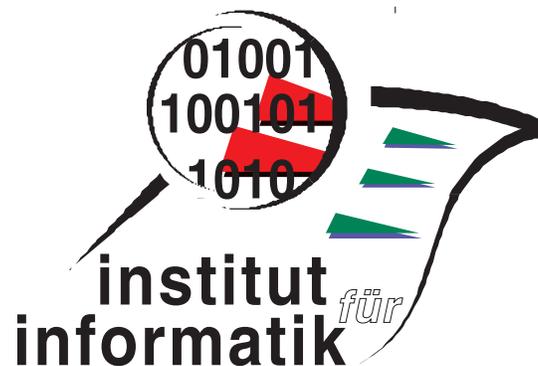# Precise Visibility Determination of Displays in Camera Images

E. Hörster, R. Lienhart, W. Kellermann and J.-Y. Bouguet

INSTITUT FÜR INFORMATIK

D-86135 AUGSBURG

PRECISE VISIBILITY DETERMINATION
OF DISPLAYS IN CAMERA IMAGES

E. Hörster, *R. Lienhart*

W. Kellermann

J.-Y. Bouguet

Chair of Mulimedia Computing
University of Augsburg
Augsburg, Germany

Chair of Multimedia Communications
and Signal Processing
University of Erlangen-Nürnberg
Erlangen, Germany

Intel Corporation
Santa Clara, CA, USA

ABSTRACT

In many novel application scenarios such as smart rooms or sensing rooms visual sensors (such as cameras) need to know which visual actuators (such as displays) are visible to them. Often only parts of a display are visible from a camera. Therefore, a novel algorithm for precise visibility determination is presented. The algorithm makes the assumption that the displays are active, i.e., they can be controlled by the application. Under these conditions the algorithm determines precisely where which parts of a display are imaged by a camera.

1. INTRODUCTION

Sensor networks and sensing rooms are promising recent research directions. Especially sensing rooms are often equipped with visual sensors (cameras) and actuators (displays). Important questions arising in this setup are (a) how to calibrate the multiple cameras, i.e. how to put them into a common coordinate system, (b) how to add the locations and orientations of displays to the common coordinate system, and (c) how to determine which parts of the displays are visible in each camera, either because the display is not fully in the field of view and/or parts are occluded by other objects [1]. In this paper we will address the last problem – the problem of determining the visibility of displays in uncalibrated cameras.

Although the topic of camera calibration from images is well investigated and a significant amount of research has been done in the last years, e.g. [2], [3], no directly related research could be found in the area of visiblitiy estimation of active displays in camera images.

The paper is organized as follows. In Section 2 the problem

is formulated, a solution overview is given and the limitations of the solution are identified. In Section 3 our algorithm is presented, before Section 4 gives implementation details and reports results. Section 5 concludes the paper with a summary and outlook.

2. PROBLEM FORMULATION AND ITS SOLUTION

Given an uncalibrated static camera and an active flat-panel display, both at unknown locations, the objective is to automatically determine the fraction of the display that is visible in the camera image. There are two possible reasons why only a fraction of a display is visible in a camera image: (1) parts of the display are simply out of the viewing area of the camera or (2) parts are occluded by at least one foreground object. We will not analyze the reason for the invisibility in our approach, but just determine which pixels are visible to the camera.

Our solution to the above mentioned problem is based on the idea to use the active flat panel display to show a known temporal pattern encoding each pixel on the screen. During playback of the temporal pattern the static camera captures an image sequence. The image sequence is then analyzed over time, whether and where some of the known temporal pattern can be detected in the captured image sequence. The detected temporal pattern directly encodes the display position to which they belong. As a result a complete list of all visible pixels is determined.

In the following we will describe the temporal pattern and its associated detector function deployed in our prototype system.

Temporal pattern: A sinusoidal signal pattern of frequency f_i is assigned to each pixel on the active display letting its grayscale values p_i oscillate between black and white (0 and

255) over time t according to:

$$p_i = \frac{255}{2}\sin(2\pi f_i t) + \frac{255}{2} \qquad (1)$$

Frequency analysis: For analysis the greyscale value of each pixel in the camera is captured over n frames to determine its oscillating frequency. It is assumed that the frames are captured with a sample frequency f_s, i.e., frames are sampled at time intervals Δ, where Δ is determined by the sampling frequency f_s of the image sequence according to $\Delta = \frac{1}{f_s}$.

For each pixel an estimation of the power spectrum is computed using the so-called *periodogram* estimate.

According to [4], given a time function $c(t)$ sampled at n equal time intervals Δ, the <u>D</u>iscrete <u>F</u>ourier <u>T</u>ransform (DFT) can be computed via

$$C_k = \sum_{l=0}^{n-1} c_l e^{2\pi i l k/n} \quad k = 0, ..., n-1 \qquad (2)$$

where $c_0 ... c_{n-1}$ denote the n samples of the function $c(t)$ ($c_l = c(l\Delta)$). The periodogram estimate of the power spectrum at $n/2+1$ different frequencies f_i is then estimated by means of:

$$\begin{cases} P(0) = P(f_0) = \frac{1}{n^2}|C_0|^2 \\ P(f_k) = \frac{1}{n^2}[|C_k|^2 + |C_{n-k}|^2] \quad k = 1, ..., (\frac{n}{2}-1) \\ P(f_c) = P(f_{n/2}) = \frac{1}{n^2}|C_{n/2}|^2 \end{cases}$$
$$(3)$$

where the frequencies f_k are defined only for zero and positive frequencies

$$f_k \equiv \frac{k}{n\Delta} = 2f_c\frac{k}{n} \qquad k = 0, ..., \frac{n}{2} \qquad (4)$$

f_c denotes hereby the Nyquist frequency. The Nyquist frequency is given by $f_c \equiv \frac{1}{2\Delta}$.

The oscillation frequency of a pixel may be determined by searching for peaks in the power spectrum. Each pixel has a maximum at $f = 0$, the DC component. Significant local maximums in the power spectrum for $f_k > 0$ mark the oscillation frequency of the pixel. If a clear pronounced maximum cannot be found, the pixel is approximately constant over time.

Limitations of the above described approach derive from different aspects. First there are physical limits: The resolution of determining the visible parts is limited (a) on the display's side by the size of each pixel and (b) on the cameras side by the size of the captured image of the display in pixels which in turn depends also on the distance between the screen and the camera.

Second mathematical limitations arise from the frequency

analysis procedure. The highest frequency that can be detected in a pixel, the Nyquist frequency, is limited by the sampling frequency of the captured sequence. The frequency resolution with which the power estimate can be calculated in the interval $[0, f_c]$ is given by Equation 4. Besides from the Nyquist frequency, the resolution depends on the number n of samples. Given n samples the power estimate may be determined by the above described periodogram estimate at $(n/2) + 1$ frequency bins.

The described limitation are crucial for the computation. Given, for instance, a display of size $1600 \cdot 1200$ pixels, the number of different frequencies f_i needed to be displayed is $1600 \cdot 1200 = 1920000$. To determine each frequency a power spectrum estimate at a minimum of 1920000 positive frequency bins has to be calculated in order to be able to distinguish all displayed frequencies. This implies at least a 3840000-point DFT. This is prohibitively expensive.

Fortunately, it usually does not make sense to assign each pixel a different frequency, as the camera resolution is usually much lower than the display resolution and additionally the display is covering only a small area in the camera images.

3. VISIBILITY ESTIMATION ALGORITHM

Due to the limitations identified in the last section the temporal pattern described above is only approximated in the actual implementation. Instead of varying each pixel in a different frequency, we divide the display into rectangular regions K_{ji} as shown in Figure 1. The rectangles result
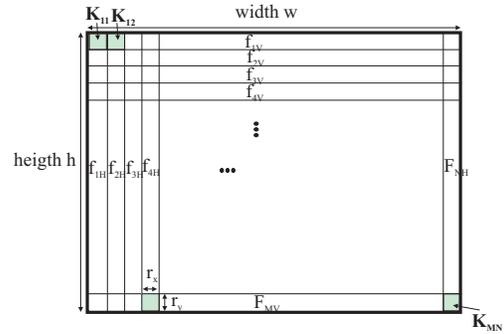

Figure 1: Approximation of the ideal pattern

from a combination of two 1D patterns displayed in sequence, the so called *horizontal* and *vertical* pattern.

The horizontal pattern is constructed by dividing the display into N regions as illustrated in Figure 2. A frequency f_i is assigned to each regions according to $f_i = f^H \frac{i}{N}, i = 1...N$. f^H denotes the highest frequency chosen to occur in the horizontal pattern. The pattern is called horizontal,

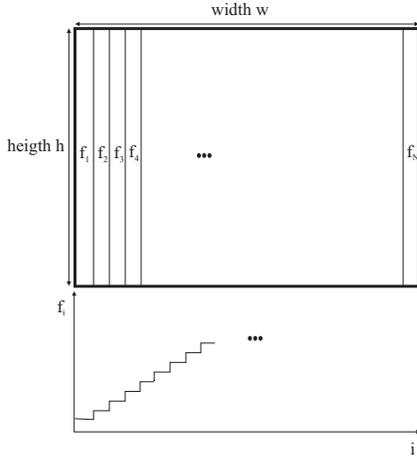because the assigned frequencies increase in horizontal direction from left to right.

Figure 2: Horizontal pattern: the greyscale value of a pixel in the region i oscillates in the assigned frequency f_i

Subsequently the so called *vertical* pattern is displayed. Analogous to the horizontal pattern, the screen is divided into M regions, where the frequency increases in vertical direction from the top to the bottom.
When both patterns are combined, the screen is divided into $M \cdot N$ rectangles K_{ji} each of dimension $r_x \times r_y$, where $r_x = \frac{w}{N}$ and $r_y = \frac{h}{M}$. Each rectangle K_{ji} is encoded by the frequencies f_{iH} and f_{jV} with which its pixels oscillate in the vertical and horizontal pattern, respectively.

We have chosen to combine two sequential 1D patterns over one simultaneous 2D pattern for the following reasons. If we choose to use a 2D pattern with the same resolution, we need to display $N \cdot M$ different frequencies, and thus we need to perform at least a $(N \cdot M) \cdot 2$-point DFT in order to estimate the power spectrum at a minimum of $N \cdot M$ frequencies (Equation 4) and thus to be able to distinguish the displayed frequencies. In the case of two sequential 1D patterns (horizontal and vertical), we can derive the same resolution by performing a $N \cdot 2$-point DFT for the captured image sequence during the horizontal pattern being displayed, and a $M \cdot 2$-point DFT for the captured image sequence during the vertical pattern being displayed. Figure 3 summarizes the algorithm.

4. EXPERIMENTAL RESULTS

The algorithm has been implemented in C++ and tested on real data. The display whose visible fraction should be detected is partially shown in Figure 4. It's screen is divided into $M = N = 40$ regions in horizontal and vertical direction. With a native resolution of 1600×1200 pixels,

- Display the horizontal pattern on the flat-panel screen and capture the horizontal image sequence.
- Extract n frames.
- Obtain a discrete time function for the grey-values of each pixel by detecting the values over n frames.
- Compute a power spectrum estimate for each pixel .
- Detect for each pixel the frequency f_l>0 at which the power is maximal. If the power value at this frequency is above a certain threshold, assign f_l to the pixel, otherwise assign 0.
- Display the vertical pattern on the flat-panel screen and capture the vertical image sequence.
- Perform all the above described steps also for the vertical sequence. Thus a frequency f_k>0 or 0 is assigned to each pixel.
- Combine the results of both sequences: Compare for each pixel, where a frequency f_l>0 and f_k>0 in the horizontal respectively vertical sequence has been detected, the detected frequencies to the known frequencies displayed in both patterns. If f_l and f_k correspond to frequencies f_i and f_j displayed in the according patterns, then the rectangle K_{ji} (see Figure 1) is identified to be visible from the camera.

Figure 3: Algorithm for determining the visible parts of a display from an uncalibrated camera

this results in a resolution of $r_x = 40$ pixels in x-direction and $r_y = 30$ pixels in y-direction. The highest horizontal and vertical frequency f^H and f^V was chosen to be 4 Hz, i.e. frequencies in different regions differ at least about 0.1 Hz. The recording sampling frequency f_s of the image sequences with the camera was chosen to be 20 frames per second (20 fps) in most experiments. Some experiments were performed at 30 fps.
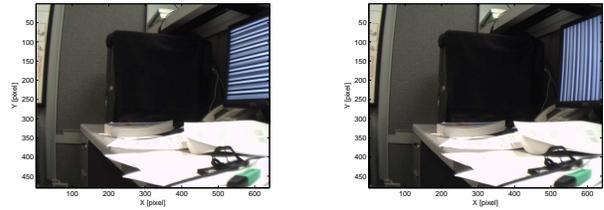

Figure 4: Sample frames of screen displaying the vertical (left) and horizontal pattern (right).

Figure 4 shows two single frames each extracted from an example image sequence: at left the screen is displaying the vertical pattern, at right it is displaying the horizontal pattern. The number of frames extracted from each captured sequence is 512. It is important to choose a power of two, as then the frequency analysis can be computed faster.

In a first stage the grey-value of each pixel in the image is recorded over n frames. The resulting discrete function of grey-values over time is shown in the right graph of Figure 5 for three different pixels each marked by a cross in the left image of Figure 5.
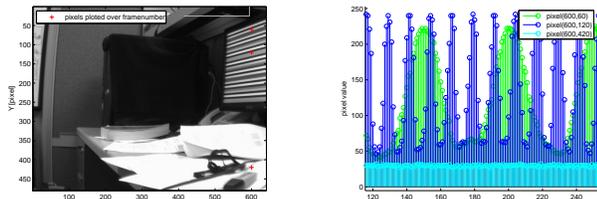The discrete Fourier transform is implemented as a Fast

Figure 5: Grey-values of the three pixels, marked left, as a discrete function of the frame number (right)

Figure 7: Examples of the camera images and the results of the algorithm

Fourier Transform (FFT). According to Equation 4, for a sampling frequency of 20 Hz the power spectrum can be computed with a resolution of approximately 0.04 Hz (the resolution for a sampling frequency of 30 Hz is approximately 0.06 Hz).
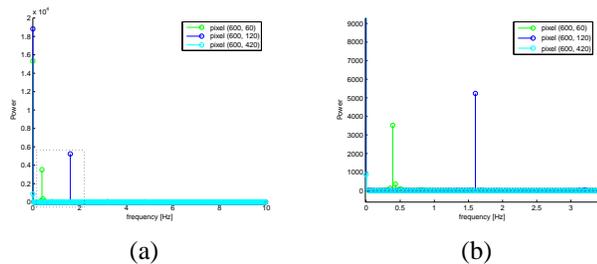

(a) (b)

Figure 6: Periodogram estimate of the power spectrum of the discrete functions shown in Figure 5(b)

The resulting power spectrum estimate for the three discrete time functions shown in Figure 5 are illustrated in Figure 6. On the left side the whole spectrum is illustrated, the right side shows in detail the part that is marked with a rectangle in the left image. All three power spectrums have their maximum value at $f = 0$, the DC component. The power spectrums of the first two pixels show also a strong local maximum for $f > 0$, whereas the third pixel whose discrete time function is not varying over time (see Figure 5) does not show a noticable maximum.

The results obtained by the algorithm for three different cases are illustrated in Figure 7. The left column shows the images from the camera's viewpoints, the right column presents the resulting visible parts of the displays in the images. In all examples the boundaries of the visible fraction of the screen are bent. This is due to the lack of camera calibration, i.e. a direct result of the camera's (radial) distortion.

5. SUMMARY AND CONCLUSIONS

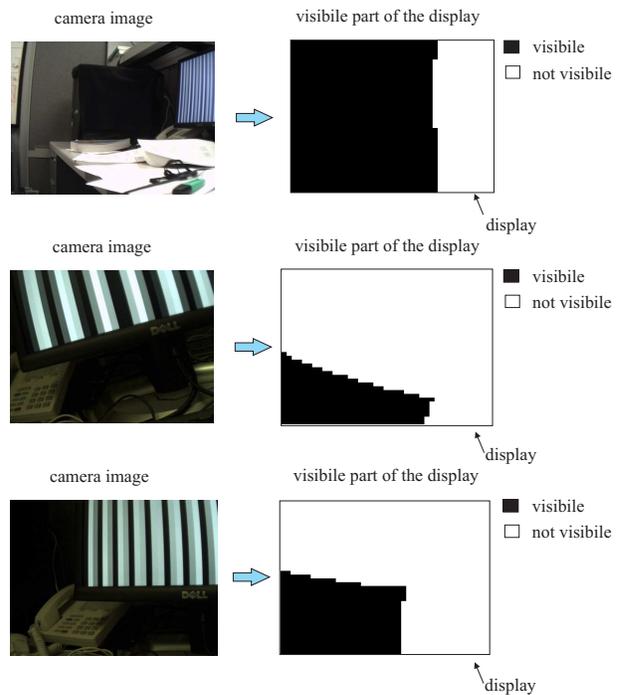In this paper we describe the problem of precisely determining the visibility of displays from uncalibrated cameras. We propose to display an appropriate temporal pattern on the display and then to analyze the respective image sequences captured by the cameras. The proposed algorithm has been implemented and extensively validated on real data. Results are also presented.

6. REFERENCES

[1] E. Hörster *Calibration of Cameras and Displays in a Distributed Computing Platform*, Diploma Thesis, University Erlangen-Nürnberg, Erlangen, Germany, 2004

[2] Z. Zhang *A Flexible New Technique for Camera Calibration*, Technical Report MSR-TR-98-71, Microsoft Research, Redmond, USA, 1998

[3] R. Hartley *An algorithm for self-calibration from several views*, Conference on Computer Vision and Pattern Recognition, pp. 482-488, 1998

[4] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery *Numerical recipes in C*, Second Edition, Cambridge Univ. Press, Cambridge, UK, 1992