Infra Red 3D Computer Mouse

Anders la Cour-Harbo°, Jakob Stoustrup

Aalborg University
Department of Control Engineering
Fredrik Bajers Vej 7C, 9220 Aalborg East, Denmark
`alc@control.auc.dk`, `jakob@control.auc.dk`

ABSTRACT
The infra red 3D mouse is a three dimensional input device to a computer. It works by determining the position of an arbitrary object (like a hand) by emitting infra red signals from a number of locations and measuring the reflected intensities. To maximize stability, robustness, and use of bandwidth, the signals are designed by means of the wavelet and the Rudin-Shapiro transforms. This also allows for easy separation of simultaneously made measurements. The measured intensities are converted to an 3D position by a neural net. The principle also applies to other applications, for instance a hand in front of monitor. We are currently constructing a prototype to test the potential of this idea.

Keywords: wavelets, Rudin-Shapiro, infra red, computer mouse, neural net

1. Introduction

The possibility of fast, robust, and inexpensive determination of the three dimensional position of a passive object is an interesting challenge, both industrially and scientifically. In contrast to a system performing 3D positioning of active objects, (such systems are well-known and widely used; one of many examples is GPS), a system for positioning of passive objects usually has to rely on signals which are emitted in the direction of and reflected by the object rather than signals emitted by the object itself. The natural consequences of this is that there is a relative large difference in the intensity of the emitted signal and the reflected, received signal. For the system to be both robust and efficient it is therefore vital to exploit its single, major advantage: The complete control and knowledge of the emitted signal.

To test a new idea for an inexpensive and robust 3D positioning system we are constructing an infra red 3D computer mouse. Many other positioning systems could be used as a test bench, but we have chosen the 3D mouse because it is cheap and relatively easy to build, has suitable real time requirements, is of some commercial interest, and it has "laboratory-friendly" dimensions. The idea is to emit a whole range of signals from various position and measure the reflected intensities. The relations between the intensities is then converted into a 3D position. The signals are infra red (IR) light, and the IR emitters and receivers are located in a box with dimensions equivalent to a thick ordinary mouse pad. An object, like a hand, can then be positioned when it is over the mouse pad. The physical design and basic principle of the 3D mouse is shown in **Fig. 1**.

The calculation of the 3D position of an object is divided into two consecutive steps. The first step is determining the relative distances by measuring the intensity of infra red light emitted in the direction of and reflected by the object. The second step is converting the high dimensional measurement data into a three dimensional position. We have developed and tested an accurate and robust method that carries out the first step. This is described in the next section, and constitutes the main part of this presentation. To perform the second step we propose to use a neural network, possibly with wavelets functions as neurons. The relative distances are highly non-linear, and preliminary simulations have indicated that a neural network with conventional functions might not suffice. The results so far is described in **Sec. 3**.

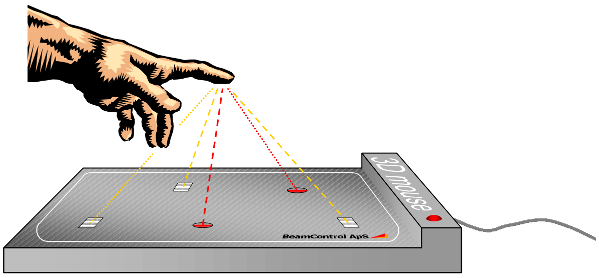

Figure 1: The physical design and basic principle of the 3D mouse. A number of infra red emitters (circles) and receivers (squares) are located in the 3D mouse under the IR transparent top cover. All necessary electronics are also located inside the 3D mouse.

2. Measuring the Relative Distances

The first step of the determination of the 3D position is measuring a number of relative distances between the object and an array of emitter and receivers located in the 3D mouse. As described, this is done by measuring the reflected intensities of emitted signals, and the challenge eventually comes down to creating an algorithm which can adaptively design these signals to be optimal under given conditions.

In an ideal theoretical scenario there are four unknown variables; the 3D position and the reflectivity of the object, and hence only four distances are required. But in a real application all measurements are subject to uncertainty, and the reflectivity might not be homogeneous across the object. The consequence is that a larger number of relative distances are required, which inevitably increases the computational complexity. At the same time the real time requirement – combined with the desire to use only inexpensive hardware – limits the available computational power.

We present in this section a measuring method that is well

This work is in part supported by the Danish Technical Science Foundation Grant no. 9701481.

Patent pending.

suited for accommodating these opposing interest by

- exploiting the full potential of any combination of emitters and receivers,
- making all the measurements simultaneously,
- being highly accurate in moderate noise conditions,
- being very robust in severe noise conditions,
- being suitable for fixed point DSP implementation.

This is achieved by using two closely related digital signal design algorithms; a "best case" and a "worst case" algorithm. The former is based on the wavelet transform (WT), while the latter is based on the Rudin-Shapiro transform (RST). They are both simple, numerically stable, and post-processing friendly, making them ideal for implementation in a fixed point DSP or a custom IC. By introducing a signal processor it becomes possible to continuously redesign the signals for improved SNR, and thereby increasing the potential for accuracy and robustness in changing and/or severe noise conditions.

Due to space limitations, we will not describe the popular WT in further detail. The RST is briefly presented in **Sec. 2.3**.

2.1. Designing the Digital Signals

The two design algorithms are based on the wavelet packet transform scheme; it is fast, numerically stable, works well in fixed point arithmetic, and has low program complexity. The best case algorithm uses the classical WT to create signals which are near-orthogonal to expected noise occurrences, while the other algorithm uses the RST to create an all-spectrum signal, which by nature has low sensitivity with respect to time and frequency located noise occurrences. The difference is essentially that the WT algorithm "searches for holes" in the current noise, while the RST algorithm spreads information in time and frequency to reduce the impact of localized disturbances. The preferred method depends on the noise conditions. If there are easy-to-find holes in the noise, the former can provide very accurate measurements. If, however, the noise is difficult to define or is changing rapidly, the latter method provides less accurate, but more robust measurements.

A good introduction to the wavelet theory from an applicational point of view is Wickerhauser [7]. A mathematically rigorous treatment of the subject is given in Daubechies [5]. For more material on Rudin-Shapiro polynomials, see Brillhart [2] and Benke [1].

2.2. The Wavelet Transform Design Method

The idea for designing a good transmission signal is the following. A designed, simple, and time localized signal is inversely wavelet packet transformed. Time localized means that in an otherwise vanishing sampled signal, there are a number of consecutive non-vanishing samples. Since these non-vanishing samples has an interpretation as coefficients of various time-frequency atoms, it is possible, by coordinating the design of the signal with the wavelet basis chosen for the inverse transformation, to create a signal with particular time and frequency properties.

After transmission the signal is forwardly transformed to reproduce the original, now noisy simple signal, and by inner product with the "clean" original signal the transmission intensity is determined. Since the original signal is completely known, it is also possible to obtain an estimated accuracy of the channel gain measurement. This is accomplished by taking inner product between the transmitted, transformed signal and a number of signals orthogonal to the original signal. If these quantities are small the transmission was most likely subject to only mild noise. This trick provides an easily calculated guideline to how much one can trust the current measurement. It is also possible to apply a number of signal restoration procedures (not further described here). These also benefit from the complete knowledge of the original signal. Note that the small number of non-vanishing coefficients of the original signal in all cases significantly reduce the amount of calculations.

Because the transform is linear and has perfect reconstruction, it is also easy to make a good sample by sample estimation of the noise, which provides valuable information on any disturbances. This makes it possible to do real time adaptation of the signal by redesigning them. The methods combines the ability of the WT to produce predefined trade-offs between time and frequency information with the freedom in design of the original signal. Thereby it is possible to adapt the method to many types of noise.

2.3. The Rudin-Shapiro Transform Design Method

The RST defined in this presentation is closely related to the remarkable Rudin-Shapiro polynomials, which were discovered in 1951 by Harold Shapiro and published in 1959 by Rudin [6]. Some of the desirable properties of the RST are inherited from these polynomials. The RS polynomials are defined as

$$\begin{aligned} P_{m+1}(z) &= P_m(z) + (-1)^{\delta_m} z^{2^m} Q_m(z), \quad P_0 = 1 \\ Q_{m+1}(z) &= P_m(z) - (-1)^{\delta_m} z^{2^m} Q_m(z), \quad Q_0 = 1 \end{aligned} \tag{1}$$

with $\delta_m \in \{0,1\}$. It immediate follows that for all $|z| = 1$

$$|P_{m+1}|^2 + |Q_{m+1}|^2 = 2|P_m|^2 + 2|Q_m|^2 = 2^{m+2}.$$

Consequently,

$$\max_\xi |P_m(e^{i\xi})| \leq \sqrt{2}\|P_m(e^{i2\pi\cdot})\|_2, \tag{2}$$

which is the single, most attractive feature of these polynomials; with (2) a certain flatness of the polynomials is guaranteed. A construction similar to (1) is found in Byrnes [3]. The flatness of the Rudin-Shapiro polynomials means that their coefficients constitute time series with a broad frequency content.

To produce such wide spectrum series we use the RST, which is really a wavelet packet Haar transform with time and frequency depended impulse response. The RST has a number of nice properties, summarized in the following theorem, which also defines the RST itself.

Theorem 1 (The Rudin-Shapiro Transform)
Define the mapping $\mathbf{H}_{j,m} : \mathbb{R}^{2^j} \mapsto \mathbb{R}^{2^j}$, $j \geq 1$, *as*

$$\begin{bmatrix} y_k \\ y_{k+2^{j-1}} \end{bmatrix} = \frac{(-1)^{mk}}{\sqrt{2}} \begin{bmatrix} 1 & (-1)^k \\ (-1)^m & -(-1)^{k+m} \end{bmatrix} \begin{bmatrix} x_{2k} \\ x_{2k+1} \end{bmatrix} \tag{3}$$

for $k = 0, \ldots, 2^{j-1} - 1$ *when mapping* $\mathbf{x}$ *to* $\mathbf{y}$. *Define the Rudin-Shapiro transform as*

$$\mathbf{H}_J \overset{\text{def}}{=} \prod_{j=1}^{J} \begin{bmatrix} \mathbf{H}_{j,0} & & 0 \\ & \ddots & \\ 0 & & \mathbf{H}_{j,2^{J-j}-1} \end{bmatrix} \tag{4}$$

Then $\mathbf{H}_J = [h_{m,n}] : \mathbb{R}^{2^J} \mapsto \mathbb{R}^{2^J}$ *is a unitary and symmetric Hadamard matrix with*

$$h_{m,n} = 2^{-J/2} \prod_{j=1}^{J} (-1)^{(m_j + n_{J-j+2})(m_{j+1} + n_{J-j+1})}, \tag{5}$$

for $m, n < 2^J$, *where* m_j *is the* j*'th bit in the binary representation of* m, *with* m_1 *LSB. Moreover,*

$$\max_\xi \Big| \sum_{m=0}^{2^J-1} h_{m,n} e^{i2\pi m\xi} \Big| \leq \sqrt{2}, \quad n = 0, \ldots, 2^J - 1. \tag{6}$$

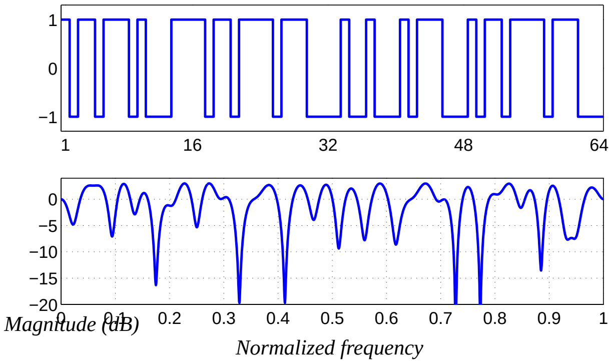

Figure 2: The top graph shows the 37th basis vector of a size 64 Rudin-Shapiro transform. Below is the amplitude of the frequency response of the basis vector.

Note that $\mathbf{H}_J$ is unitary because $\mathbf{H}_{j,m}$ is unitary, that (6) is equivalent to (2), and that it follows from (5) that $\mathbf{H}_J$ is a symmetric Hadamard matrix. We omit the complete proof, which will appear later in the ph.d. thesis of the first author.

The theorem shows that the RST constitutes an orthonormal basis, which not only consists of uniformly scaled ± 1, but also, due to (6), has a remarkable frequency response. In **Fig. 2** is an example of such a basis element and its frequency response.

The brute force procedure for decomposing a vector in the RST basis is to multiply it with the inverse of $\mathbf{H}_J$. However, the theorem shows that, not only is $\mathbf{H}_J$ its own inverse, there is also a wavelet packet like scheme, given by (3) and (4), which performs the matrix multiplication with complexity $O(J \log J)$. Note that if the 2×2 matrix in (3) was replaced by the first of

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad \begin{bmatrix} 1 & (-1)^k \\ 1 & -(-1)^k \end{bmatrix},$$

then $\mathbf{H}_J$ would be the all-scale wavelet packet Haar transform. If replaced by the second matrix, the rows of $\mathbf{H}_J$ would be the coefficients of the Rudin-Shapiro polynomials defined in (1). Of these only the Haar transform is symmetric.

The RST based algorithm is applied in much the same way as the WT method. A designed, simple signal is transformed prior to transmission, yielding an all-spectrum signal. Upon transmission the signal is transformed again, this time producing the original, simple signal with noise. The post-processing is equivalent to that of the WT method.

3. Determining the 3D Position

The second step of the 3D positioning is converting the relative distances into a 3D position by means of some mapping from a high dimensional space to $\mathbb{R}^3$. This mapping defines the interpretation of any quality of measurement; perfect, good, bad, as well as completely wrong, and must do so in a very short time due to the real time requirement. The mapping should fulfill the following prioritized requirements:

1. It works well for good measurements,
2. there is a reasonable relation between error in measurements and error in 3D position,
3. it has low computational complexity,
4. it has low dynamic range in computations,
5. it is easily adaptable in real time,

Since the measurements are expected to be good most of the time, the primary concern is that the mapping does well in this case, and the second requirement ensures that a small decrease in accuracy does not result in too large deviations in the 3D position.

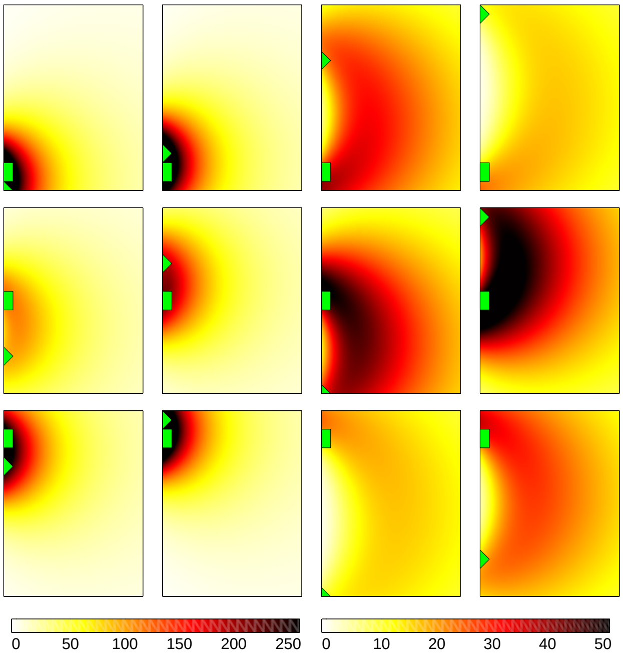

Figure 3: The simulated reflection intensities. The triangle is the emitter, and the square the receiver. The first two columns have the same color scale, and so does the last two. The axes limits are the same as in the other figures.

3.1. The Properties of the Neural Net

There are various ways of constructing this mapping ranging from completely theoretical, geometrical consideration to purely ad hoc methods. We have chosen a middle-road approach by using a neural net. On the one hand this offers a systematic and fairly well-described way of defining and describing the desired mapping, and at the other hand requires a lot of guessing and testing. Moreover, a neural net has the potential of fulfilling the above requirements, as is shown in the following.

In this particular framework there are two ways of using a neural net; as a classifier and as a function approximation. The former is useful if only one of a few possible positions are needed instead of the actual position. This applies, for instance, when pointing at icons on a monitor. In this presentation only the function approximation network is investigated, being the most interesting type in the case of the 3D mouse.

We have chosen to use a radial basis function network, because it is well-suited for function approximation, plus it requires only a relatively limited amount of training. For a more detailed description on radial basis function network, see Chen et. al [4].

3.2. Simulating the 3D Positioning

To simulate the 3D positioning by a neural net it is necessary with measurement data from an array of emitters and receivers. Acquiring almost error-free data by means of a real electronic setup is difficult; it requires extensive work and expensive equipment. So, instead a model is used to produce the measurements. This is a rather complex and computationally heavy model, which simulates the reflection of a sphere at any given position by means of a ray-tracing like procedure. We have chosen to present a 2D simulation, since this is much more suited for visual interpretation (plus it is significantly less computationally complex).

The setup consists of 4 emitters and 3 receivers located along a line (equivalent to location in a plane in the 3D mouse). The simulated measurements are shown in **Fig. 3** along with the location of the emitters and receivers. Note that the overall inten-

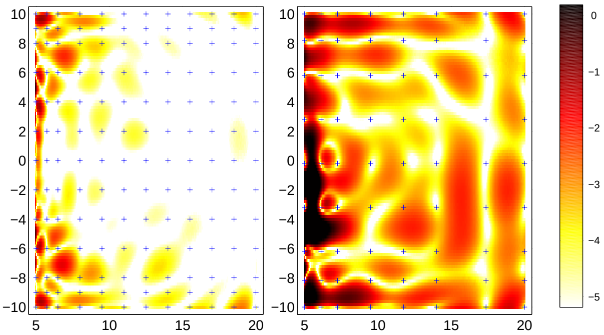

Figure 4: The error in distance (Euclidean norm) between true 2D positions and 2D positions simulated by a neural net. The crosses mark the training points. The left set of training points yielded 82 neurons, while the right gave 57 neurons. The color scale is $\log_2$.

sity depends heavily on the distance between emitter and receiver. The size of each measurement set is 16×20 units (which might be interpreted as centimeters). The idea is now to use a neural net to map the 12 dimensional measurements to a 3D position. The network is constructed by repeatedly adding neurons (which in this case are functions on the form Ae^{-t^2}) until the MSE between the true and simulated 2D position in a set of training points is below a threshold. While this procedure specifically reduces the error in certain points, the goal is to have a good approximation in all 2D points. The former does not necessarily imply the latter, as is shown in **Fig. 4**. Here two sets of training points are used, and the MSE of all the training points are 0.3. In between the training points, however, there is no control of the the error, which can easily become quite large. But adding training points in places with large error will inevitably also increase the number of neurons (to meet the MSE threshold requirement). One obvious goal is to have as few neurons as possible, but it is equally important that the neural net is not too sensitive to noise. To test this (on the neural net with 82 neurons) the net has predicted the 2D position based on 12 dimensional measurement data with added Gaussian noise. This is shown in **Fig. 5**. Since the added noise have the same variance all over the 16×20 plane, while the measurements are varying in amplitude (as seen in **Fig. 3**), the SNR varies somewhat. Although the weaker noise is typical for laboratory tests, the stronger noise is not uncommon. This figure shows one major weakness of the neural net; the large sensitivity to even Gaussian noise (the scale is $\log_2$, so the predicted positions are useless).

The problem is that although the ‘clean’ measurements are 12 dimensional, they constitute a 3 dimensional submanifold since they are originally mapped from $\mathbb{R}^3$. If a 12 dimensional measurement is too far from this embedded submanifold the prediction made by the neural net becomes arbitrary and hence useless. We have two potential solutions to this; the neural net could be trained for erroneous data as well, or some projection onto the 3D submanifold could be applied in the 12 dimensional data space. We have tested the former idea with positive result, but a significantly larger number of neurons is needed, since an even more complicated structure than the 3D submanifold is approximated. The latter solution is somewhat more complicated, because it requires a fine-gridded non-Euclidean multi dimensional structure (consisting of splines, for instance) of the 3D submanifold in order to facilitate computation of a numerical projection, as an analytical projection is not feasible. We have not yet tested this idea.

4. Conclusion

We have proposed to construct a 3D positioning system by combining a computational simple and robust algorithm for measuring relative distances with a neural network. The former is based on the wavelet and the Rudin-Shapiro transforms, which are well suited for moderate and severe noise conditions, respectively. They both have low program and computational complexities in addition to good numerical properties, making them suitable for low cost hardware implementation. The neural network presents some difficulties, and some work still remains to be done in this context. One untested possibility is using wavelets as basis function in the neural net; we expect this to reduce the number of neurons because of the greater flexibility of wavelets compared to Gaussian functions. However, the preliminary simulations indicate that this method for converting relative distances into 3D positions has significant potentials.

For the commercial aspects of these methods, please refer to `www.beamcontrol.com`.

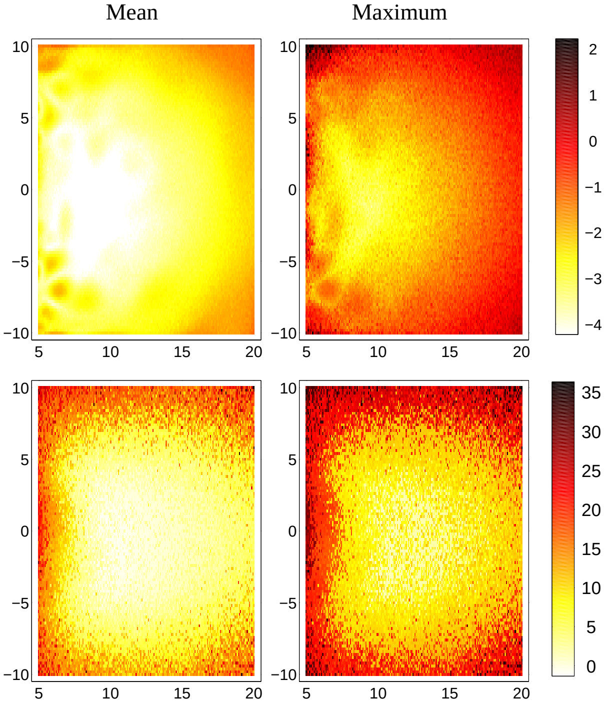

Figure 5: The mean and maximum distance error for 200 instances of Gaussian noise (SNR range from 50 to 25 dB in the uppermost row, and 30 to 5 dB in the lower row). The color scale is $\log_2$.

References

[1] G. Benke. Generalized Rudin-Shapiro systems. *J. Fourier Anal. Appl.*, 1(1):87–101, 1994.

[2] J. Brillhart. On the Rudin-Shapiro polynomials. *Duke Math. J.*, 40:335–353, 1973.

[3] J. S. Byrnes. Quadrature Mirror Filters, Low Crest Factor Arrays, Functions Achieving Optimal Uncertainty Principle Bounds, and Complete Orthonormal Sequences – A Unified Approach. *App. and Comp. Harm. Anal.*, 1:261–266, 1994.

[4] S. Chen, C.F.N. Cowan, and P.M. Grant. Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. *IEEE Trans. Neural Net.*, 2(2):302–309, march 1991.

[5] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.

[6] W. Rudin. Some theorems on Fourier coefficients. *Proc. Amer. Math. Soc.*, 10:855–859, 1959.

[7] M.V. Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*. A K Peters, May 1994.