3-D Sensing Using Smart Sensors: Compact Efficient Sensor Processing

by

Matthew A. Clapp

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

October, 2005

Abstract

Artificial devices that are active and participate in the real world are dependent on their sensor systems to provide a snapshot of the environment around them. This vital connection between external reality and internal representation can determine the entire character of how a machine interacts with its surroundings. For systems whose only purpose is sensing, this is the beginning and end of their abilities. Even for many varieties of robotic applications, it can be the limiting factor on performance. Visualizing the environment in the three spatial dimensions is commonly the most desired capability.

In all but the most specialized applications, often sensor processing methods attempt to convert specific sensor signals into the most generic digital representation possible, followed by digital processing at some location likely distant from the sensor itself. Digital processing allows some flexibility in processing specifications, but comes with a price. System resources can be wasted by trying to most details of the raw data in digital form for long-distance communication. The entire process involved for "packaging" sensor data for digital computation risks losing

essential qualities available at the sensor. It also incurs penalties in performance and power efficiency.

I demonstrate in this thesis two sensor processing systems following a contrasting philosophy from such traditional methods. Sensor processing is carried out as near to the physical sensors as possible, capturing essential features immediately. By also concentrating on specific representations of data designed to be most useful, efficiency is the result. A visual imager and sonar array system are shown to and yet use little power. They carry out their functions more in a superior way for certain applications than presently existing alternatives. In some cases, they enable new applications not previously possible because of extremely optimized performance.

Advisor: Prof. Ralph Etienne-Cummings

Readers: Prof. Andreas G. Andreou and Prof. Virantha N. Ekanayake

Acknowledgements

In this section, I humbly thank my family, friends and colleagues for their long-standing (and vital) support:

My family for emotional (and occasionally financial) support: Dad, Karen, Alice, Michelle.

My extended family from the East Coast for making my stay on this coast feel like home: grandparents, aunts, uncles, cousins.

For everything and more: (especialmente) Bea.

Housemate of six years for mirth, (in)sanity, wine and song: Alexis.

Friends & Colleagues at JHU supplying equal amounts of mental aid and humorous relief: Viktor Gruev (co-founder of Etienne lab), Swati (goddess of serenity), Francesco, Kat, Jacob, Ralf, Udayan, Zi, Mike Chi, Philippe, Dave Goldberg, Eugenio, Alyssa, Mark Martin, Roman, Milutin, Hi-Hi & Loosy.

Professors and mentors: Ralph, Andreas, Gert, and Virantha.

Pure friends and non-colleagues from JHU: Manuz, Tamar, Caline, Silvio, Soo.

Friends from Caltech slightly ahead of my academic schedule: Noah & Van-

dana, Yun, Zack.

Friends from Minnesota to remind me of my original intellectual goofiness: Dan, Ryan, Amy.

Friends from California who flattered me by wanting me to come back: Andrew Hsu, Samantha, Rob Chapman, Renny.

I also thank my sponsors: This work was supported by the a DARPA research grant, and the NSF ERC on Computer Integrated Surgical Systems and Technology at Johns Hopkins University.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

One of the defining aspects of human history has been the desire and skill for creating better and better tools. The first tools facilitated manual labor, increasing mechanical advantage and multiplying the effectiveness of physical work. As basic needs were met, people had both the time and the need to create more sophisticated tools. A cycle developed in which new technology both enabled and created a desire for even better tools and technology. Today, the relatively advanced technology we use in our daily lives is the result of this long lineage of innovation based on human inventiveness.

As technology matured along with society and culture, our demands evolved from purely mechanical to computational. Analogous to their mechanical coun-

terparts, the first computing machines multiplied mental labor by helping people calculate mathematical problems many times faster than an unassisted person could. Their speed and capabilities increased at an amazing rate. Computers have now improved to the point where they can facilitate many different kinds of non-physical tasks such as writing, drawing, and communication, where their mathematical underpinnings are not obvious or unseen altogether. They are increasingly seen as something more than mathematical engines.

As our perceptions of computers become more and more general, it is tempting to imagine that the next logical evolution of these machines is to "think" for themselves. Currently computers behave much as any other tool–they may multiply our effort, but only in the sense that they follow our strict commands and specific instructions for their behavior. Inducing a machine to behave more autonomously as an assistant and less as a tool would allow more work to be accomplished with less oversight from people.

If intelligent devices are going to become more active, increasing their ability to assist humans, they need a proper awareness of the world they are going to operate in. This awareness cannot simply be recording video or audio, or any other type of raw data. It must include capturing features of the world in a more meaningful way. Data must not merely be gathered, but it must be understood. Understanding the world begins with a machine's interface to the world, its sensors. Its sensors must supply clues to help it navigate and be effective in its environment, not just

record it.

A machine able to interact with the world in real-time needs to be able to receive useful features from its sensors in real-time. Here, both the rate of processing and the delay, or latency, from stimulus to response is important. A mobile machine possessing untethered freedom to interact with its environment must also accomplish such tasks in an energy-efficient way. A strategy addressing all of these concerns, enabling high-throughput, low latency, and low-power consumption, is to embed some low-level pre-processing at the sensor level itself. Unintelligent sensors which merely forward raw data to a distant processor necessarily make a trade-off between the detail of the data sent and the power and complexity involved with communication of the data. Processing carried out close to the sensor can take advantage of immediate, low-level access to more of the sensor data without much of the costs of long-distance transport of that data. Such "smart" sensors extract useful features from raw sensory data before passing the result to higher-level processing. The quickness with which they are able to translate reality to actionable data can even enable any number of applications which were previously not possible, especially those involving feedback loops requiring no latency and high update rates. For these applications, energy efficiency is an additional bonus.

Applications for sensors in isolated environments are inherently resource-limited. For a battery-operated wireless sensor, both low-power operation and efficient

coding of information to be transmitted are vitally important. Because wireless data transmission is often a significant use of power, smaller bandwidth is often synonymous with significantly lower system power consumption. Ideally, isolated sensors could be placed over a wide area without fixed (wired) infrastructure for power or data transmission. For the dream of distributed sensing to become a reality, engineers will have to break their dependence on standard method of digitizing analog data as close to the source as possible, at higher and higher bandwidths to capture more nuances of the signal. The power needed to run a fast ADC and the huge amount of data it produces may be too costly, especially if much of that data is redundant or useless. For tiny data collectors, with miniaturized power supplies and miniaturized transmitting antennas, efficiency is everything.

An autonomous device equipped with only one type of sensor may be able to operate in a limited or well-controlled environment, but its versatility will be limited. There are numerous situations in which multiple methods of sensing the environment are vital to its effectiveness or even survival. A person can avoid a hot stove because they can feel the heat from it, even though it may not be visibly red-hot. An assembly-line worker can feel that a part is not smooth even if by sight it is not visibly rough. A car in need of repair may make a distinctive sound despite the fact that all gauges indicate normal operation. Most animals possess multiple senses merely to survive in their environment. Evolution thus argues strongly that complementary ways of sensing one's environment are very useful even for very

simple animals.

1.2 Principle Objectives

The sensory systems I describe in this work are designed to provide functional information about the environment to either a mobile robot, a distributed sensor system, or a feedback-based system fast information about the world. Beyond merely sensing, these systems provide a first layer of processing from sensory input information. They extract and output useful features about the world, which may be used directly by a central processing unit. These features include locations of fixed or moving objects in three spatial dimensions. To facilitate more general imaging of the environment, features may also describe useful parts of fixtures such as walls or terrain to facilitate navigation.

Target applications include resource-limited situations such as guidance for a small mobile robot, an isolated sensor in a distributed sensing system, or various biomedical applications. A number of design goals can be derived from these situations. A fundamental goal is to keep the sensor and processing system as physically small as possible. Specifically, sensor and sensor processing should be kept close to the level of a single-chip solution. The physical sensor may necessarily be off-chip, and minimal supporting circuitry may also be necessary. But my aim is to provide a compact package capable of extracting features from raw sensor data

locally. I avoid a situation where raw sensor data must be prepared for long-term communication before it can be processed.

Secondly, power efficiency is essential for miniature remote or highly mobile applications. Most current methods for storing energy are limited by small power-to-mass ratios. Roaming robots or isolated sensors with sizes measured in the tens of centimeters or less must be frugal with power usage in order to last long enough to be effective without continually recharging.

The speed of sensing, beyond mere throughput, is also vital. The target applications benefit greatly from instantaneous computation that is not forced to make its way through a long computational pipeline. Thus, design solutions will minimize latency as well as increase throughput.

Every sensor technology is more useful in certain settings than in others. In order to enable my devices to be generally effective in the widest possible circumstances, I will create multiple sensor types which can be used together. This prevents the "blind spots" of any one sensor technology from causing a mobile or isolated device to be without any information about its environment. In addition, information from multiple sensor types can provide a richer, more multi-dimensional picture of the world. In this work I describe an optical imager and a sonar array designed to be used in conjunction with one another. The multi-dimensional combined view yields the three spatial dimensions and time.

Purely optical or purely sonar systems have been used in the past for sensing

in 3-D. A sonar system is an active system, sending out the sound energy that it eventually receives reflected off of distant objects. Because it provides its own "illumination", it has the advantage of not depending on ambient conditions for sensing. It is also excellent for detecting the range of objects, because the range of an object from the sensor can be directly calculated by how long the reflection takes to return from the object. This contrasts with optical methods for depth determination, which may require more than one camera and intensive processing. By using multiple microphones in a centimeter-scale small array of sonar receivers, the angle from which an ultrasonic ping is returning to the array can also be detected, allowing for a full 3-D polar representation if two arrays are used.

However, an active sensing system is able to be detected because of its emissions. In instances where stealth is important, this is a major disadvantage. A passive system such as an imager receives ambient light already present in the scene. Thus it can be used in situations such as surveillance where secrecy is important. Passive systems may save energy over active systems because they are not responsible for sending out the energy they receive. Finally, an optical imager in a camera can also provide improved spatial resolution for the 2-D projection of a scene without depth.

It can be seen that a sonar array and optical imager have different and complementary strengths and weaknesses. I will show that by using my particular processing implementations, and by allowing for the combination of the two, it is

possible to extract a more energy-efficient and more accurate representation of the world.

1.3 Past Approaches to 3-D Sensing

1.3.1 Optical only

Traditional optical methods for sensing 3-D have included using two cameras with stereo processing, and optical flow methods.

Probably the most popular method in the mobile robotics community has been the use of stereo cameras with massive processing on general-purpose computers. In fact, the high level of interest of the robotics community in stereo vision underscores the importance placed on detecting the world in 3-D, and the need to detect objects and features of the world, not just images. The typical implementation involved two regular CCD video cameras, communicating full frames to a Personal Computer (PC) which would perform fairly compute-intensive stereo processing. Because of the physical size of the processing implementations, the size of the mobile robots involved are on the order of meters, larger than the target applications of this thesis.

The basic idea of extracting depth from stereo cameras depends on the concept of parallax. If the same scene viewed from two separate side-by-side positions, objects nearer to the cameras will show more apparent side-to-side displacement

between the two views than more distant objects will. By computing the relative displacement of the same object or point as it appears from two different viewpoints, the depth of that object or point can be calculated. The geometry of this problem is fairly straightforward, as is the math required to deduce depth from displacement.

The most difficult and processing-intensive part of the operation usually is the need to determine which two points in the two images correspond. Typically, a candidate patch of one image is compared with a search neighborhood in the other image, and the closest match is attempted to be found. This search involves many iterations, comparing each patch of one image to multiple patches on another image. Classical matching techniques require additions, multiplications, and possibly even square root functions to be applied to data on each iteration.

Various non-idealities of the images may frustrate this process. First of all, exposure errors, including and gain and offset of exposure may exist between the two images. These are collectively termed radiometric errors, and can cause certain matching algorithms (such as Sum of Absolute Differences [SAD], or Sum of Squared Differences [SSD]) to give erroneous results. In addition, spherical distortion from optics and imperfect alignment of the direction of the cameras can make the geometrical calculations much more complicated than the ideal. One solution is to carefully mechanically align both cameras. Practically, it is usually necessary to use image processing to warp every raw camera image to a non-distorted

version before depth processing. These images are termed rectified images. Use of rectified images greatly simplifies the depth processing, but rectification itself represents non-negligible computational effort.

Other methods for block matching exist that require less arithmetic computation yet still allow for good matching performance. These "non-parametric" methods such as rank and census matching require only comparisons within each patch to determine which pixels are lighter or darker, and storage of the resulting comparisons. Only the ordering of brightnesses matters, which also tends to prevent radiometric errors from affecting the matching algorithms. At the same time, these particular matching algorithms allow the matching process to be carried out by comparing at most 1 bit per pixel in each patch. The data involved is thus far less than the full brightness values required by more classical methods. Because of the reduced computation, matching for stereopsis has been implemented in systems composed of FPGAs only [2, 3]. This is a marked reduction in hardware sophistication for this task compared with the advanced processors mentioned earlier. However, it is important to note that in the above references, the boards were still fairly complex, containing approximately 68,000 equivalent gates and large amounts of fast cache memory, dedicated only to stereo processing. In addition, the actual tests of their computation used rectified images because the hardware involved does not carry out the image warping on each image necessary to prepare them for matching. This can still be a formidable use of processor resources if full video

rate (30fps) 3-D vision is desired.

Nowadays, as standard microprocessors used in PCs become more advanced, most implementations of stereo processing use these general-purpose chips instead of dedicated programmable hardware. Modern PC microprocessors include units using Single Instruction Multiple Data (SIMD) techniques, often used for the multimedia extensions. Such processors are optimized for video processing in addition to their general computational tasks. The advanced processing comes with a cost, however. Pentium II, III, and IV chips used for this purpose can consume anywhere from 40W to 100W while carrying out computation.

In summary, there have been many systems that implement 3-D vision from stereo imagers, with some decent performance. Unfortunately, the computation is costly, and successful systems involve much hardware, and consume more power than small applications can afford.

1.3.2 Sonar only (Traditional Sonar and Sonar Arrays)

For a variety of applications, sonar is a popular choice for remote object sensing and remote imaging. Underwater sensing of ships at sea is possibly the oldest and most well-known large-scale deployment of sonar [4]. In the past few decades, sonar technology has increasingly come to be used on smaller scales. Anyone who has seen a sonogram of a developing fetus is familiar with sonar's widespread use in medical imaging [5]. Diagnostic ultrasonic waves are medically benign to

the patient, involving no radiation exposure as X-rays do. In addition, ultrasonic imaging is both vastly more portable and lower-cost than other medical imaging technologies such as Magnetic Resonance Imaging (MRI). Such advantages are also of prime importance to designers of mobile robots. For these applications, the low cost and low processing requirements of a sonar ranging system are clear advantages over other popular methods used to sense the environment. One alternative, laser range-finding, is usually seen as more accurate—but its common drawbacks include higher cost, higher power consumption, and faster processing requirements.

A mobile robot built to navigate its environment with sonar sensors is usually faced with a few well-known challenges. The first is the attenuation of ultrasonic sound frequencies in air, which is considerably more than the attenuation in water. Greater attenuation causes the ultrasonic signal to fall off more quickly, and reduces the strength of received reflections from distant objects. The second issue facing designers of ultrasonic systems is the wide angle of sensitivity of most ultrasonic microphones. In an ideal sense, it is convenient to think of sound waves emanating out in a straight line and reflecting back on the same straight line. In reality, however, most real-world ultrasonic sources have beams which encompass a relatively wide area. Reflections occur from almost every object and reflect specularly in multiple directions, because all objects appear as smooth surfaces to long-wavelength (up to 8.5mm in air) near-audible sound waves. Ultrasonic

microphones receptive over a large angle receive all of these reflections. A naive processor thinking that a sonar return can only be from an object directly in front of the sonar sensor will give erroneous, puzzling results in this all-too-common scenario [6]. In the mobile robot community, much effort has been devoted to solving this problem. More and more complicated systems involving multiple large sonar transducers, complicated heuristic or statistical processing, and large processing units have been described in the literature [7, 8, 9, 10]. The trend in these systems is to increase the amount of information available to the processing units by using multiple sensors instead of individual ones. The logical conclusion to such work is array processing, a mature, decades-old field which has already been effectively used to enhance the reception of both electromagnetic and sound waves [11].

Medical and ship-based sonar systems have been using classical sonar phased arrays for some time now. Phased array systems use multiple receivers separated in space to receive a signal from a single source. The angle of the incoming signal relative to the array determines the phase of the signal at each point in space. Simple processing can cause the array to be most selective to a particular incoming signal direction. For instance, summing the response of all sensors in a regularly-spaced linear array will cause the largest output response when all sensor signals are in-phase with one another. This will only happen when the signal source is normal to the line of the array. Variations on this concept include adding particular phase delays before summation to select a certain angle. More complicated

digital processing allows for the possibility of selecting multiple angles, and also of lessening the response to a certain direction to eliminate an interfering signal. Naturally these techniques can also be extended to a two-dimensional array to sense input bearing of both altitude and azimuth, which with range yield a full 3-D polar representation.

Most ultrasonic phased arrays are sampled-time systems. For traditional delay-sum methods, the sampling rate is not based on the frequency of the ultrasonic carrier signal, as one might assume. Because relative phase information is the basis of the whole process, sampling of the signals has to be carried out at a time resolution greater than the smallest inter-sensor signal delay possible—often making the sampling rate much higher than the Nyquist rate for the carrier signal. In [12], a delay-sum system using analog sampled-time delays and analog summation is presented. While potentially compact and low-power, details of the implementation's power consumption were not presented. It is sensitive to only one particular direction at a time, requiring multiple pulses or implementations to scan an angular space. Also, the minimum time resolution demonstrated in the reference was 1μs. For the microphone array spacing used in this research, time resolution on the order of 100ns would be necessary to achieve sufficient angular resolution for its needs. A high sampling rate is even more costly for array processing accomplished with Analog-to-Digital Converters (ADCs) and Digital Signal Processors (DSPs), the form of many modern approaches [11]. Digitization of all members of

the array can represent a large use of power and a significant flow of data to the DSP, which for real-time operation must keep up with the incoming flow of data and carry out array processing. Even for sonar operating at near-audio frequencies, implementing the classical digital phased array approach can thus require significant processing speeds, power consumption, and cost. And shrinking the array further would require proportionally higher clock speeds. To achieve a truly integrated MEMS-type array 5 times smaller than ours would necessitate each sensor's sampling rate to exceed 50MSamples/s in a clocked digital system, requiring much more power and complexity for data conversion and processing than is warranted for one sensor array. Quadrature sampling schemes can allow successful array processing with sensor sampling rates close to the Nyquist frequency of the carrier [13, 14]. While these methods obtain more relaxed timing requirements, they require demodulation of the incoming signals, and depend on increased voltage precision in the data conversion step. In addition, digital implementation in the above references utilize fairly fast digital hardware. In these works no power consumption data is presented, but power usage for the types of FPGAs employed can easily approach 1W each.

Such design costs may be too high for a number of applications, such as object detection for mobile robots.

1.4 My approach (Resource-Limited Processing)

I have described the traditional method of sensor processing as high-speed digitization close to the sensor, with high-speed data transmission followed by high-speed digital processing. Rejecting this method as too costly in system resources of bandwidth and power necessitates an alternative approach. The processing systems of this work all utilize mixed-signal VLSI techniques to accomplish more efficient processing where standard digital systems are not the most advantageous use of system resources.

1.4.1 Make Use of Analog Input Data Close to the Source

At the start of the sensor-processing signal chain, sensor data is by definition analog. Even if ostensibly a sensor provides two-state data, the quantifiable variable of interest is almost always analog. Some examples of analog data from two-state systems include: the time between states, the number of transitions of states, or the duty cycle the two states. The sensors of interest in this work are more obviously analog. Both photo-diodes and microphones produce continuous output currents or voltages proportional to incident light intensity or air pressure, respectively.

Digitization of analog data is a resource-expensive process. Extracting the useful components of the analog data and eliminating useless or redundant data be-

fore digitization reduces the amount of analog data needed to be digitized. For applications in which a feasible analog processing solution exists, it makes sense to remove the digital middleman and make use directly of the analog data from the sensor. And as I shall argue, sometimes a continuous-time or continuous-value processing methodology can yield better efficiency for a given application anyway.

Besides processing efficiency, being able to make use of raw signals present at the sensor eliminates the need to transmit a high-fidelity representation of the raw data to a remote processing unit. There is often a very clear trade-off between power used and bandwidth in any communications channel, and this is especially so for wireless communication. Even driving precision signals off-chip requires non-negligible power, whether the signals are represented by analog voltages or currents, or digital lines. The more processing taken care of "in-house" within the sensor processing chip, the better.

1.4.2 Asynchronous Continuous-Time Circuits

With certain processing schemes described in this thesis, it is possible that during most of the time that waveforms are being input, no events relevant to processing will be occurring. At the same time, when something does happen, such important signal changes often need to be processed with great precision. The sonar processing unit is a good example of such a situation. While pulses can last as long as 100μs, when pulse edges occur they should be processed them with

sub-100ns resolution.

Synchronous or time-sampled approaches to data processing sample a digital or analog signal at regular intervals. The interval must be set small enough to satisfy the worst-case time precision requirements. If the precision requirements are strict, sampling will occur at a very fast rate. If events of interest are rare, however, most of the time this sampling will be for no reason at all and simply be a waste of resources. The biggest waste of resources in this case is usually power: from the analog buffers copying precise analog sampled signals, or from the switching of digital circuitry where dynamic power consumption is dominant.

For these applications, a more efficient solution is to use continuous-time circuits that rely on circuit precision for processing precision, instead of the period of a sampling clock. This asynchronous approach can require more design ingenuity to implement successful circuits, but this is an acceptable trade-off for significant power savings.

1.4.3 Parallelism

The sensor processing that has been implemented also makes use of a collection of primitive processing units instead of one or two powerful units. By aggregating many independent units, the errors of each can be averaged, reducing the entire system error.

For my imager, if a few of the computational pixels give faulty response, it

will affect the total position result of the moving object only slightly. Because my sonar processing method uses a nine-element sensor sonar array, errors in precision between any two sensors are averaged and reduced when all nine sensors are considered. This is a powerful advantage over simple triangulation using two sensors.

The parallelism in these sensor systems has been designed as an integral part. By making this philosophy fundamental to the design and not merely an afterthought, the effectiveness and speed of the implementation is greatly enhanced.

1.4.4 Application-Specific Processing

Finally, I make use of dedicated processing at the sensor as a final optimization. Generic sensors with generic processors are obviously the most flexible means to accomplish computational tasks. However, for many of the envisioned applications, the cost of power and complexity using totally general-purpose hardware is too great. As shall be seen, there exist compelling uses for sensor feedback where the performance of generic solutions is not even adequate. The overhead involved with digitizing sensor data, and packaging it properly to be compatible with multi-purpose processors can necessitate significant delays if near-instantaneous reaction to a perceived event is desired.

The loss of flexibility in computation does not mean that the final use of the data is predetermined, however. The products of of the designed sensor systems aim

to be useful for many applications, even though computed by dedicated means. The feature representation endeavors to be basic enough to allow many types of higher-level uses, but rich enough to allow certain lower-level applications to be able to efficiently and speedily use information about their environment. This level of computation is a step up from raw sensor data, and should also aid the combination of multiple sensors by a processor attempting to coordinate disparate sensors' views of the world.

1.5 Original Contributions

My computational imager both acts as a general camera to produce a faithful image of the scene it is viewing, and a computational imager outputting the position of a moving object in the scene. It employs a novel system using multiple pixel types to carry out computation efficiently as well as producing high-quality images. These two versions of the same scene are automatically registered with each other, because they are taken from the same focal plane. While many computational imagers use non-integrating current-mode pixels for computation, this often results in poor image quality. Likewise, for processing based on integrated pixels and time-sampled, continuous-time processing is difficult or impossible. By using two types of pixels, an expedient method is presented to allow for both strategies to be used simultaneously. Much of the processing for the motion-tracking

functionality is accomplished on the edges of the focal plane, allowing the pixel implementation to be compact to preserve fill factor. Additionally, computation of the centroid of the moving object is done in real-time, immediately updating as pixels in the imager indicate temporal changes. Because position is always being updated, readout of the position can be read at any time, although waiting for a longer period of time allows for a greater population of pixels to be averaged in the centroid position and greater accuracy. This position tracker has been comfortably run in a mode where it outputs 180positions/sec., or 6 times the normal full-motion video rate. The latency involved is also virtually none, at maximum less than the time between position readings, or 1/180s. Because of the extremely fast update rate and no latency, it is possible to use this imager output in a feedback loop, something possible only at very slow speeds with 30fps conventional imagers and normal pipelined digitization and computation of image data. All of this computation and imaging is done on the same chip, using a very small amount of power to accomplish both, 2.6mW. [15, 16, 17, 18, 19, 20]

The sonar array described takes input from nine microphones and outputs bearing angle data in addition to the traditional range data usually performed by sonar systems. The processing involved is a novel use of spatiotemporal frequency filtering, used previously in vision systems but apparently never for processing on an ultrasonic array. Conceptually, a wavefront moving across the sonar array is viewed in this system as a velocity. The spatiotemporal frequency filters

isolate the energy of this velocity in spatial and temporal frequency space to deduce the magnitude. Once the velocity of the wavefront is determined, the translation to incident angle of the sonar return is straightforward. This method is uses continuous-time computation, unlike most array processing schemes which use sampled-time techniques. Because time is not sampled, there is no global clock. This provides some advantages. In a very small baseline system such as the one I am pursuing, the clock of sampled-time system would need to be very fast in order to precisely capture inter-sensor timings to enable bearing detection. A fast clock necessitates faster analog circuits which use more power, and faster digital circuits which use more power. The power used in such conventional designs is merely to accommodate a clock based on the worst-case precision necessary, and represents a waste of power. In my system, circuit time precision and a novel method of computation is used instead of a power-wasting global clock. The spatiotemporal frequency filter also inherently performs an averaging over all nine sensors, reducing the effect of timing errors from any one sensor. My mixed-signal implementation of the algorithm is also very efficient, using approximately 1mW of power to discriminate with 1° accuracy over an angular range of approximately 170°. In addition, the system allows for a much more plausible migration path to even smaller sensor arrays, such as could be produced with Micro-Electro-Mechanical Systems (MEMS), than can be envisioned for sampled-time systems, where scaled clock frequencies would need to be hundreds of megahertz or even gigahertz. This

would represent an unacceptable use of computational resources for just one sensor. [21, 22, 23, 24, 25]

The sensor systems also provide processing frameworks that are amenable to be combined in a future sensor fusion process. The structure of processing in each is not monolithic, but rather uses discrete levels of processing from raw sensor data to highest-level output. The internal processing structure is constructed in a way which could allow an external data fusion processor to be able to access lower-level but still processed representations of the sensors' worldview. By making the architecture of these sensor processors modular in this way, a combination of the two sensors for a full 3-D measure of the world using two separate modalities should be much more readily achieved.

1.6 Thesis Overview

We first describe two sensors designed to provide sensory data as well as carry out selected preprocessing to extract vital features from the data.

The first sensor described is a CMOS imager array. The imager is meant to be operated behind the lens of a camera as any imager chip for a digital still or video camera would be. The architecture of the imager is presented, with details of the computations preformed. The theory of operation is studied, including how different noise components affect operation. Next, characterizations of various

circuits are shown, culminating with the performance of the entire system. Finally, applications of the imager are described where the advantages of efficient motion processing are either highly advantageous or necessary.

The next sensor/processing combination described is an ultrasonic microphone array, and associated array processing electronics. I detail some background of typical Time-of-Flight sonar for finding range to objects first, including attempts to make sense of confusing data resulting from range-only data. These ambiguities lead directly to the need for 2-D or 3-D sonar sensing. Next is a review of spatiotemporal frequency filtering theory, before moving on to its application to a sonar array. The particular VLSI implementation of spatiotemporal frequency filtering is described, followed by characterizations of all sub-circuits and processing blocks. Finally, the system as a whole is characterized while listening to an ultrasonic beacon and then in a full sonar test involving sound reflected from objects. Finally, applications to make use of this new advanced sonar system are outlined, showing the advantages of using this new processing system.

The balance of the thesis provides an overall picture of the advantages, organized by target application, of these new sensing systems. After showing the benefits of using each system individually, the future work necessary to bring both sensors together into a hybrid system is sketched. In the course of the description of the necessary work to fuse the two sensors' data together, that the processing structure of each sensor system lends itself well to sensor fusion is also explored.

Chapter 2

Imaging and Motion Centroid Tracking Array: Theory and System Architecture

2.1 Introduction

Lately, there has been much interest in CMOS imagers as an alternative to CCD arrays [26]. Specifically, Active Pixel Sensors have been utilized along with Correlated Double Sampling circuits to produce charge-integrating imagers approaching CCD image quality [27]. In addition to image quality, the most important advantages of using CMOS technology include: lower power-consumption, lower fabrication costs, and the ability to integrate processing tasks on the same chip as

the imaging array. The advantages of system integration have been thoroughly examined, and are the basis of the "System-On-A-Chip" approach [27]. Specifically, integration enables greater flexibility of processing, simplifies interconnects between imaging and processing blocks, and further reduces costs by assembling all functionality onto one die.

Many CMOS imager designs use these advantages simply to make a cheaper, more efficient substitute for a CCD array and scanning circuitry. The goal of the designed chip described below, however, is to extract information from a scene with a degree of speed and power efficiency that wouldn't be practical with physical or logical separation of the imaging and processing blocks.

The combination of imaging and computation is not a new concept. Numerous designs for imagers have been created in the past with computation and imaging combined on the same die [28]. The most notable examples of this design philosophy carry out computation on the focal plane itself. Most of these designs use current-mode pixels, employing photodiodes or phototransistors to produce a current which is an instantaneous function of the incident light [29]. The use of current-mode pixels facilitates continuous-time processing, and enables the use of space-efficient analog computation circuits. Unfortunately, when image quality is important, current-mode pixels suffer a worse noise performance than their charge-integrating cousins [30]. Integration serves to average out such noise fluctuations. Furthermore, capacitors are more easily matched than the transconduc-

tance or threshold voltages of transistors. The disadvantage of an integration method is that it produces discrete-time data that is not as compatible with many space-friendly analog processing circuits.

While this situation seems to create an irreconcilable choice between image quality and focal-plane computation, there is a conceptually simple way to achieve both. I have created a chip that uses two distinct types of pixels in the same array in order to reap the unique benefits of each. One pixel-type is a standard Active Pixel Sensor for lower-noise imaging, and the other is custom-designed to help compute the centroid position of a moving object. The two pixel types are interleaved in the array to ensure automatic registration of imaging tasks. The imager is called the Dual Pixel Centroid Tracker imager or DPCT imager.

The tracking of object position has traditionally been important for military applications, and much of the related literature of the past couple of decades is devoted to this area [31]. As imagers and processors have become cheaper, however, their potential uses have also broadened [32, 33]. Machine vision for mobile robots, automation of security camera tasks, image stabilization for medical applications, and other motor control applications have now become practical. For mobile platforms such as robots, low power is very important. Devoting processing to objects in its environment that are moving is a logical choice for a robot seeking to apply "attention" where it is most useful. Likewise, running images of a static scene through power-hungry processing is a waste of resources. Letting a

low-level sensor alert the robot to image areas that require further processing is an efficient solution to these issues. For systems that include the centroid calculation in a feedback loop, high speed and low latency are vital. Latency is an especially important issue for image stabilization and mechanical feedback systems. They demand quick response or risk becoming oscillatory or simply ineffective. In retinal microsurgery, the tremor of a surgeon's hand is a hazard because precision on the order of microns is required. Likewise, tremors of the subject's eye can cause a blood vessel being operated on to exhibit movement. Optical tracking of both the surgeon's instrument and objects in the operating field coupled with mechanical compensation of surgical instruments could make the resulting operation jitter-free. Finally, offloading the task of computing a centroid frees memory and processing time, which can be used for higher-level tasks such as image segmentation or for other jobs not related to image processing. In a mass-produced item such as a security camera, centroid-tracking capability built into the sensor could enable a useful but menial task (detecting intruders) to be automated with little extra cost. With direct access to the image plane, my centroid-tracking chip described below can carry out the same tasks with much less energy and space than a separate processor could.

There have been other approaches to centroid location and tracking in the past. The most common approach, mentioned previously, involves communicating all pixels from the imager block to a processing block, usually a digital processor, be-

fore computing the centroid [34]. This approach not only involves the computation time of the processing block, but also the time it takes to read every pixel out of the array itself into memory. For a system with a digital processor, part of the communication process is analog-to-digital conversion–costly in terms of time, chip area, and power. If pixels are communicated out of the array one at a time as is customary for most imagers, the time involved scales as $O(n^2)$ with the length of a side of the array. As resolution increases, this additional time involved in moving pixel information can become considerable. A 2Mpixel sensor with a 10MHz pixel clock takes 200ms to output every pixel—costly in terms of latency. The pixel clock speed can be increased, but this again costs more chip area and/or power.

In contrast, the DPCT imager described in this thesis finds the centroid of moving objects within the scene using focal-plane computation. The centroid-tracking pixels in the array include circuitry for both photoreception and preliminary processing. Basic processing at the pixel level allows the elimination of more complex processing later. Edge circuitry receives simple binary data from these pixels in parallel along shared row and column lines. With one edge circuit cell per row or column, the size of the edge circuitry scales linearly with the size of the edge of the array. Digitization of pixel analog values is not necessary. There is also no need for a separate storage elements, because the centroid is computed instantaneously from the pixels themselves. Since the computation is carried out in parallel, the speed of centroid circuitry is nearly constant over all imager sizes, de-

pending mainly on the circuit reset frequency. The most significant scaling effect is an increase of line capacitance for row and column shared lines proportional to the square root of the total pixels. This is an issue related primarily to rise and fall times of these signals, and will not dramatically affect computation speed.

Some prior centroid-tracker designs use the focal-plane for fast, low-power processing, but they compute the centroid of the whole scene based on the brightness of every pixel in the imager [29, 35]. While useful for tracking a bright spot in a scene, these imagers are unable to discriminate between objects of interest and the background. The DPCT imager chooses to find moving objects as "interesting," and computes the position of these objects independently of the appearance of the background. This rule is a simple way increase the number of situations that allow the centroid-tracking system to yield meaningful data.

Of course, in addition to tracking moving objects, one would actually like to be able to view the objects and the scene itself as a regular image. High fidelity images are important to applications such as medical imaging, where accurate visual information is important to human beings who will be viewing them. Good image quality is also important for computer vision applications that use the centroid-tracking capability of the chip as a first pre-processing step before more complicated algorithms that need the full image information. For these uses, current-mode photoreceptors such as those used in the computation pixels would produce an image that is too noisy. Instead, Active Pixel Sensors are used for imaging. They

are tiled with the computation pixels so that the centroid position reported has a very direct and strict mapping to the pixel position in the image. There is no calibration necessary as would be required for an optical setup involving two separate cameras.

The rest of the chapter will be devoted to analyzing APS and centroid-computation subsystems to derive expected behavior and performance. The following chapter examines results from the actual operation of the chip. I will note and explain how the results are consistent or differ from the characteristics derived in my analysis.

2.2 System Overview

The DPCT imager consists of two subsystems, the APS imager subsystem, and the centroid-tracking subsystem. The APS array operates as an imager for obtaining real-time images of what the chip sees. The centroid tracker computes the location of moving targets within the scene. Each can be operated independently of the other. In this design, no resources are shared between the two except for the focal plane itself.

Figure 2.1 shows the floor plan of the array and edge circuitry. The pixel for centroid computation is exactly twice as long on each side as the APS, to facilitate tiling. Pixels in the same row are of the same type, and the array rows alternate between centroid-localization pixels and APS. Due to the difference in size of the

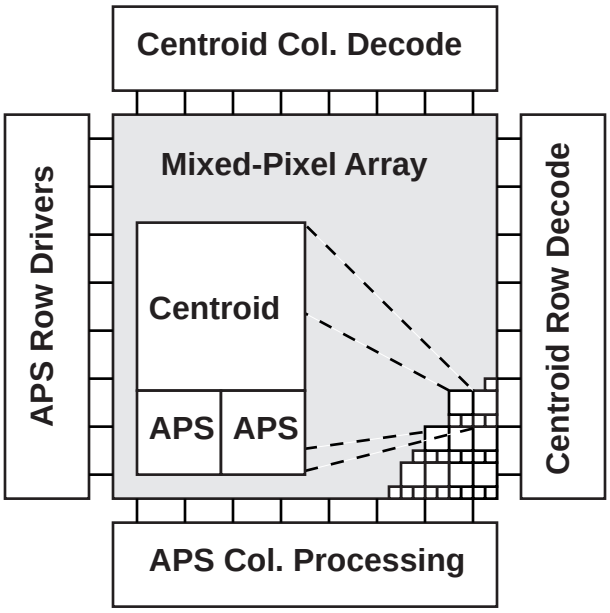


Figure 2.1: System-level view of chip

pixels, APS rows are 120 pixels across while each centroid row contains 60 pixels. The non-uniform arrangement of pixels was motivated by a desire to increase APS resolution for better image quality. Unfortunately, this decision was directly responsible for APS matching and performance problems, which will be discussed in more detail below. Digital lines are run along rows to keep the digital switching transients for one style of pixels from coupling to signals in the other. The chip was fabricated in a standard analog 0.5μm 1P3M CMOS process by Agilent.

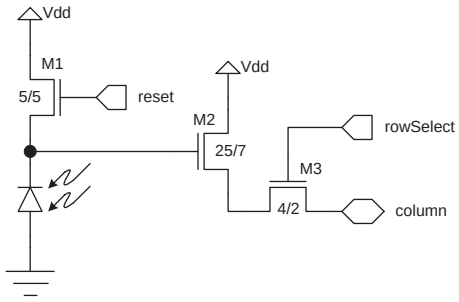


Figure 2.2: APS pixel schematic

2.2.1 APS Imaging Subsystem

The design of the APS pixel follows the same basic three-transistor/one-photodiode design pioneered by Fossum et al [26]. See Figure 2.2. Included are a reset transistor, an output transistor, and a transistor select switch to address the pixel. The structure is simple, with no provision for electronic shuttering. It is optimized primarily for density and secondly for fill factor. All transistors in the pixel are NMOS, to reduce the area of the pixel.

The row circuitry is comprised of two cyclical shift register chains: one for row reset signals, and the other for row select signals. Each row of pixels receives reset and row-select signals from one stage of each chain. Clocking these shift registers advances their bit pattern forward by one row, starting readout of the next row. The reset shift register can be pre-loaded with blocks of ones and zeros in a flexible way, allowing integration time for each row to be specified as a modifiable fraction of the total frame time. This can be viewed as a "rolling shutter." In addition, there

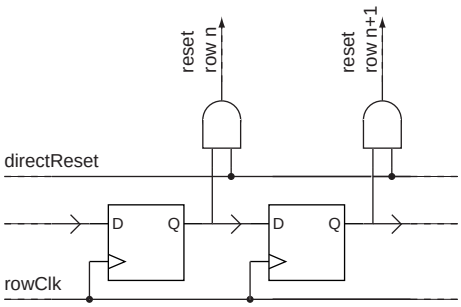


Figure 2.3: APS row circuitry.

is circuitry on the reset lines to facilitate reset timing on a shorter time scale than one row clock. A separate global signal, "directReset" in Figure 2.3, is AND'ed with each row's signal from the shift register. Using this signal, integration can be stopped and reset initiated in the middle of one row's output cycle. This is especially important to facilitate the operation of the Correlated Double Sampling system, as described below. It also allows for easy prototyping, allowing a simple method to examine the operation of one row's pixels without running the entire array.

Each column of pixels has its own dedicated processing circuitry. The column circuitry starts with its most important block, the Correlated Double Sampling (CDS) circuit [36, 37]. The job of the CDS circuit is to subtract the reset voltage from the signal voltage, ensuring that only the difference between the two is measured instead of the absolute signal level itself. This drastically reduces offset errors in readout. It also compensates for noise and different reset voltage levels resulting

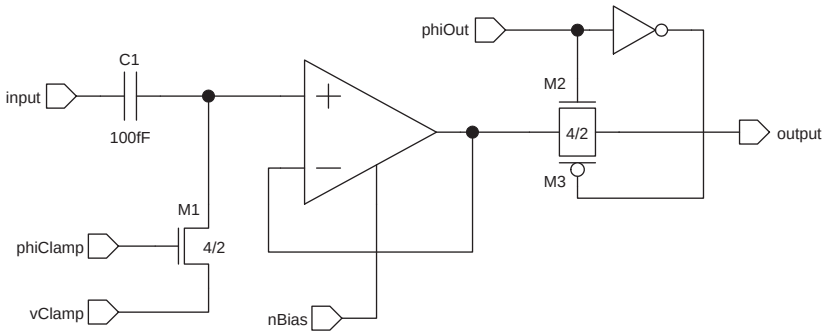


Figure 2.4: CDS, column buffer, and output switching circuit.

from different light intensities during reset. The CDS is implemented as a simple switched capacitor circuit, producing a single-ended output voltage. A fully differential circuit would have exhibited more immunity to power supply ripple and interference from other signals. However, since these reasons weren't compelling for the application, it was decided to stay with a simpler method in order to shorten design time and minimize the area of the layout. Making efficient use of space is especially important for this circuit because it is used in each column. A simple 7-transistor (diffamp and inverter) opamp with Miller compensation in a unity-gain configuration follows the CDS circuit for output buffering. Finally, the end of the column circuit employs yet another shift register chain to sequentially activate the switches that output one column voltage at a time to the single-pin output. Figure 2.4 shows a schematic of the CDS circuit.

2.2.2 Centroid Tracking Subsystem

The basic functionality of this subsystem is the computation of the centroid of all pixels whose brightness levels vary with time. This approximates finding the centroid of a moving object. A moving object will at least cause pixels at its edges to change, (in the case of a solid-colored object, for example,) and at most many pixels within the object's image will also change if it contains details or texture. The centroid of time-varying pixels in both images will be close to the center of the object. This scheme works most accurately for small objects, simply because no points on a small object are very far from the centroid. The uncertainty in pixel activity-detection will thus cause a smaller possible error in centroid position computation. In the particular design used in the DPCT, implementation details necessitated that only an increase in brightness is detected; the reasons for this modification are explained below. With this alteration, moving bright objects on a dark background should theoretically be tracked by their leading edge, and dark objects on a bright background by their trailing edge. This may cause more deviation from the true centroid in some situations. However, most real-world objects do have texture and are not solid-colored. In these situations, many pixels inside the outline of the object will be activated besides the outline pixels, lessening the impact of ignoring intensity decreases. The output of this subsystem is a set of two voltages: one for the x position, and one for the y position.

The method employed to detect pixel brightness changes can be thought of as a

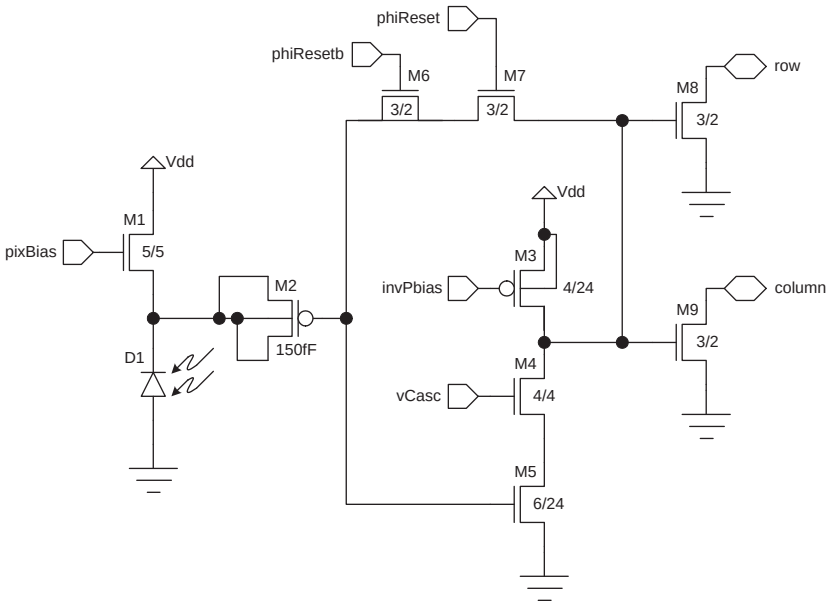


Figure 2.5: Centroid-tracking pixel schematic.

simplified form of an address event representation imager [38, 39, 40]. That is, the only output from each pixel is a digital event: the assertion of that pixel's row and column. Edge circuitry then processes the activated rows and columns to find the centroid. Moving the more-complicated processing to the edges of the array keeps the pixel size smaller and helps to increase fill factor for the motion-sensitive pixels.

2.2.2.1 Centroid Pixel

The pixel itself starts with a photodiode, which is biased by an NMOS transistor with its gate voltage fixed. (See Figure 2.5.) The voltage at the source of this load NMOSFET will be proportional to either the logarithm or the square root of

the incident light intensity, depending on whether the photodiode current operates the NMOS in the sub-threshold or above-threshold region respectively. Since the goal is to detect a relative change in brightness, the circuit is designed to be sensitive to the same multiplicative change in photocurrent at any absolute brightness level. Such contrast sensitive photodetectors have also been observed in biological visual systems [41]. The logarithmic transfer function of the sub-threshold transistor translates a multiplicative increase or decrease in photocurrent into an additive increase or decrease in output voltage, simplifying the task for the next stage of the pixel. The square root function does not share this property exactly, but it has a similar curve and approximates a logarithm. Fortunately, for most light levels the pixels of the DPCT chip operate in the subthreshold region. Light intensities of over 10mW/cm^2 are required to generate over 1nA of photocurrent, and in practice even extremely bright light conditions do not exceed 1mW/cm^2 at the photosensor. The photosensitive voltage is AC-coupled to the rest of the pixel through a PMOS capacitor with the well tied to the drain and source. The rest of the pixel consists of a resettable comparator circuit, implemented using a biased CMOS inverter, and a feedback switch. The inverter includes a cascode transistor to enhance gain.

Operation of the pixel starts with reset of the comparator block within the pixel. The inverter feedback switch is closed, input is made equal to output, and the inverter settles at its switching voltage. At this time the voltage difference between

the photodiode cathode and inverter input voltage is stored across the PMOS capacitor. The PMOScap is held in inversion, since the inverter reset voltage is significantly lower than the photodiode voltage. When the switch is opened, the inverter goes into open-loop operation. As the light level on the photodiode increases, the voltage on its cathode will decrease. Since the input to the inverter circuit is floating, (high impedance,) its voltage will now track the voltage on the photodiode, offset by the voltage across the capacitor itself. When the voltage decreases by a given amount ΔV corresponding to a given factor increase in photocurrent, the inverter will trip and its output will go high. If light on the pixel decreases, however, no event will be signaled because the inverter will move even farther away from its switching threshold.

The amount of change in incident light necessary to trigger the pixel after reset is released depends on the setting of V_{pbias}, and the amount of light incident during reset. The setting of V_{pbias} will set the reset voltage of the inverter. To maximize the gain of the inverter and to save power, subthreshold-level currents are used in the inverter. By equating the drain currents of the PMOS and NMOS transistors we can write

$$I_{0N} e^{\frac{V_{\mathrm{out}} \kappa_n}{U_t}} = I_{DN} = I_{DP} = I_{0P} e^{\frac{(V_{dd} - V_{\mathrm{pbias}}) \kappa_P}{U_t}} \tag{2.1}$$

where U_t is the thermal voltage, κ_N and κ_P are subthreshold slope factors, and I_{0N} and I_{0P} are process-dependent and gate geometry-dependent factors re-

lating9+6 fundamental currents of the subthreshold transistors. The equation for output reset voltage is thus:

$$V_{\text{invreset}} = \frac{U_t}{\kappa_N} \ln\left(\frac{I_{0P}}{I_{0N}}\right) + \frac{\kappa_P}{\kappa_N}\left(V_{dd} - V_{\text{pbias}}\right) \tag{2.2}$$

From this equation and the fact that I_{0P} is significantly less than I_{0N}, we can see that $V_{\text{invreset}} < V_{dd} - V_{\text{pbias}}$. The difference between V_{invreset} and V_T of the NMOS row and column pull-down transistors determines the initial output ΔV necessary to cause the NMOS pull-down transistors to conduct and signal a change.

$$\Delta V_{\text{out}} = V_{TN} - V_{\text{invreset}} \tag{2.3}$$

Dividing this ΔV_{out} by the gain of the inverter yields the ΔV_{in} necessary to sufficiently move the output.

$$\Delta V_{\text{in}} = \frac{\Delta V_{\text{out}}}{A_{\text{inv}}} \tag{2.4}$$

where the gain of the inverter with subthreshold drain current is:

$$A_{\text{inv}} = \frac{-g_{m5}}{\left(\frac{g_{ds4}}{g_{m4}}\right) g_{ds5} + g_{ds3}} \tag{2.5}$$

$$A_{\text{inv}} = \left(\frac{-\kappa_N}{U_t}\right)\left(\frac{1}{\frac{U_t}{V_{0N4}\kappa_N}\frac{1}{V_{0N5}} + \frac{1}{V_{0P3}}}\right) \approx \frac{-\kappa_N V_{0P3}}{U_t} \tag{2.6}$$

yielding

$$\Delta V_{\text{in}} = -\frac{\Delta V_{\text{out}} U_t}{\kappa V_{0P}} \tag{2.7}$$

or

$$\Delta V_{\text{in}} = -\frac{\left(V_{TN} - V_{\text{invreset}}\right) U_t}{\kappa V_{0P}} \tag{2.8}$$

V_{0N} and V_{0P} are the early voltages of the NMOS and PMOS transistors in subthreshold. Because of the cascode transistor, the g_{ds} of transistor M5 no longer makes a significant contribution to the gain, and can be neglected in the final expression. Note that ΔV_{in} is negative, due to the negative gain of the inverter. This equation describes the basic operation of the pixel. More details, including the effect of the coupling of the switching voltages, will follow later in the Analysis section.

The inverter drives two NMOS pull-down transistors, attached to the particular row and column lines associated with the pixel. These lines are set up in a wired-OR configuration, with weak PMOS pull-up transistors on the edges of the array. Switches can disconnect the array from the edge circuitry to avoid current draw during reset.

2.2.2.2 Centroid Edge Circuitry

To compute the 2-D centroid, the centroid of the activated rows and activated columns are separately computed in order to arrive at a final (x, y) coordinate. A center-of-mass algorithm is employed, resulting in sub-pixel precision.

The following is a description of the edge circuitry operation for the column edge specifically, keeping in mind that row edge operation works identically. Refer to Figure 2.6 for the schematic of this module. The edge of the centroid subsystem receives a series of column outputs corresponding to each column of the cen-

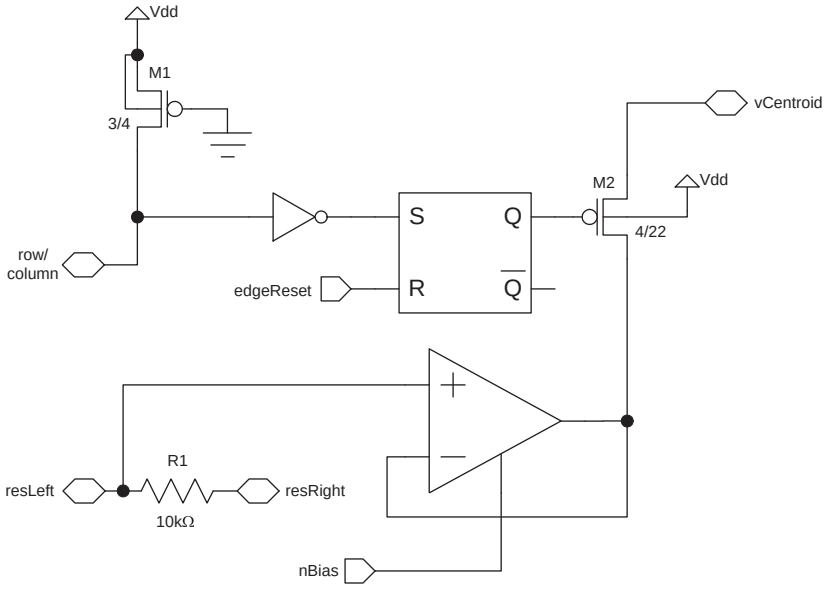


Figure 2.6: One segment of the centroid edge circuitry.

troid pixel array. Columns containing pixels that have experienced an increase in their brightness will show up as logic low. The center-of-mass calculation computes a weighted average of every activated column using the column position as weight. For example, if only columns 20 and 21 have been activated, the result of the center-of-mass calculation would be 20.5. This example illustrates sub-column position precision. The position weights are represented as a set of voltages from a resistive ladder voltage divider with as many taps as there are columns. These voltages are buffered using simple 5-transistor differential amplifiers. A column with a low (activated) output will first set an SR flip-flop, locking it high until the flip-flop is reset with an externally provided reset signal. The outputs of the SR

flip-flops turn on weak PMOS transistors operating in the ohmic region, which connect the column weight voltages to the centroid output node. The PMOS transistors have a width/length ratio of 4/22, and are turned on by lowering their gate fully to ground. All active columns will have their weight voltages connected to this common node through the PMOS pseudo-resistors, and this network of voltages interconnected through identical pseudo-resistors computes the average of all voltages connected. The output voltage is thus the center of mass value of all active columns.

In the ideal case, all activated PMOS resistors would be in the linear region, so that V_{ds} has a linear relation to the current flowing, approximating a true resistor. For a PMOS to be in the linear region, the condition $-V_{ds} < -V_{gs} + V_T$ must hold. Equivalently, $V_{ds} > V_{gs} - V_T$. It must be true that $V_{gs} < 0 - V_{\text{laddermin}}$, where $V_{\text{laddermin}}$ is the low voltage of the resistive ladder. Therefore, the sufficient condition for linear operation can be expressed as:

$$V_{ds} > -V_{\text{laddermin}} - V_T(V_s) \tag{2.9}$$

This equation also indicates the dependence of the threshold voltage on the source voltage, because of the bulk effect. In addition to this, it can be seen that $V_{ds} > V_{\text{laddermin}} - V_{\text{laddermax}}$ must always be true, because the minimum drain voltage possible is the low voltage of the resistive ladder, and the maximum is $V_{\text{laddermax}}$,

the maximum value of the voltage ladder. The value of $V_{\text{laddermin}}$ that satisfies

$$V_{\text{laddermin}} - V_{\text{laddermax}} > -V_{\text{laddermin}} - V_T \tag{2.10}$$

or

$$V_{\text{laddermin}} > \frac{V_{\text{laddermax}} - V_T}{2} \tag{2.11}$$

will cause V_{ds} to absolutely satisfy the condition for operating in the linear region. For a typical V_{TP} of -1V and $V_{\text{laddermax}}$ of 2.8V, this gives the low voltage of the resistive ladder as 1.95V to guarantee all PMOS transistors operate in the linear region.

In the normal mode of operation, the low voltage of the resistive ladder used is 1.8V, and the high ladder voltage is 2.8V. In the worst case it is possible that sometimes a PMOS transistor will not be operating in the linear region, and hence dominate the averaging operation by its higher conductance. In practice, however, moving objects are localized. As long as there is only a single moving object in the scene, the activated rows and columns will be in close proximity to one another. Hence, the V_{ds} voltages between the reference voltages and their average will stay small enough to keep each PMOS operating in the linear region.

Notice that each pixel position factors into the center of mass calculation with equal weight. Because the region of interest is defined as everywhere pixel light intensity has changed, it is necessary to assume that every point has a weight of 0 or 1. It is possible to imagine other functions, such as one that would weight each pixel by how much its light intensity has changed. However, it is unclear that this

is even a desirable metric. Therefore, it is assumed that the meaningful algorithm for the system is a binary condition: change or no change.

Also of note is the fact that this circuit does not consider the number of pixels activated in a column or row. It gives every column or row the same weight independent of the number of activated pixels. Instead of noting the actual centroid of the pixels that are activated, it detects the centroid of a rectangular box coincident with the edges of the region of activated pixels. This was chiefly an implementation-related optimization. It is much easier for the edge circuitry to note activity/non-activity than to include how much activity a certain row or column contains. For most objects, the centroid of a coincident rectangular box is a good approximation of their true centroid. The main drawback of this modified centroid is that single outlying pixels are given as much weight as those which are clustered together. Thus, false activity registering on one pixel gives that pixel's row and column the same weight in centroid calculations as rows and columns containing many pixels where the real object's image lies. This is a regrettable disadvantage, but justified because of the much simplified implementation of the current scheme.

2.3 Analysis

2.3.1 APS Analysis

The APS subsystem is composed of two main units: The APS pixel, and the edge circuitry that both controls the pixels and helps process output analog data.

The APS pixel used here is a well-proven 3T design, (reset, amplifier, and read-out switch transistors,) which has been well analyzed in other papers [27, 42]. The main characteristics of the specific implementation of the DPCT imager is summarized below. Figure 2.2 shows a schematic of the APS pixel, and Figure 2.4 shows the column readout circuit.

2.3.1.1 APS Pixel Linearity, Gain, and Sensitivity

The gain of the pixel, starting from incident light and progressing to an output voltage, is a function of only a few circuit elements. The first is the integrating capacitance on the photodiode node, 94.2fF. This sets the conversion gain of the photodiode at $1.70\mu\mathrm{V}/\mathrm{e}^{-}$. Following the input capacitance is the gain of the pixel gate-source amplifier. The output impedance of the column current sources and g_{mb} set this gain at 0.77. The switched-capacitor of the CDS circuit ideally performs subtraction of voltages with a gain of one. Leakage currents and coupling in the switches will introduce error, but because this is not a gain error, gain will be assumed to be one for the purposes of this analysis. Following the switched-

capacitor is a two-stage opamp connected in unity-gain configuration. As such, its actual gain is more like $A/1+A$, where A is the gain of the opamp. In the DPCT design, A is around 15,000, which makes the gain of the buffer configuration virtually unity.

At this point, the total gain of the system is $1.31\mu\text{V}/\text{e}^-$. To translate this to a useful figure, it is necessary to change the units of e^- to $(\mu\text{W}/\text{cm}^2)\cdot\text{s}$ by assuming a quantum efficiency of 20%, a wavelength of 600nm, and noting that the photodiode area is $30.87\mu\text{m}^2$. The resulting gain equating voltage to light intensity and integration time is $244\mu\text{V}/((\mu\text{W}/\text{cm}^2)\cdot\text{ms})$.

2.3.1.2 APS Noise

The noise of the APS system begins with the photodiode itself: Dark current shot noise, and photon shot noise give:

$$\left\langle v_{\text{photon}}^2 \right\rangle = \frac{I_{\text{photo}}\Delta t_{\text{reset}}}{C_{\text{pdiode}}^2} q \tag{2.12}$$

$$\left\langle v_{\text{dark}}^2 \right\rangle = \frac{I_{\text{dark}}\Delta t_{\text{reset}}}{C_{\text{pdiode}}^2} q \tag{2.13}$$

With $I_{\text{photo}} + I_{\text{dark}} \approx 2\text{pA}$, $\Delta t_{\text{reset}} = 926\mu\text{s}$, and the 94.2fF capacitance of the photodiode node, total noise from current through the photodiode comes to about 33.4e-9V^2, or $183\mu\text{V}_{rms}$.

Reset noise is calculated to be 210μV_{rms} from the following basic equation:

$$\left\langle v^2_{\text{pixreset}} \right\rangle = \frac{kT}{C_{\text{pdiode}}} \tag{2.14}$$

As Tian [43] notes, this noise figure is only appropriate for reset times long enough for the photodiode to reach a steady state during reset. The usual mode of operation for the APS array involves a reset as long as a full row readout time (926μs.) It turns out that this is long enough for the pixel to reach steady-state reset for moderately high light levels. However, for lower light levels, the non-steady-state noise energy figure of

$$\left\langle v^2_{\text{pixreset2}} \right\rangle \approx \frac{kT}{2C_{\text{pdiode}}} \tag{2.15}$$

should hold. This corresponds to a voltage of 148μV_{rms}. This thermal noise gives the fundamental noise floor of the images, regardless of matching.

Following the reset pixel noise, there exists noise associated with the output follower pixel amplifier and the input to the CDS circuit. During the clamping of the CDS capacitor, the noise can be modeled as kT/C noise with the CDS capacitor of 100fF. This noise amounts to:

$$\left\langle v^2_{\text{cdsclamp}} \right\rangle = \frac{kT}{C_{CDS}} \tag{2.16}$$

or 41.43e-9V^2.

After the output side of the CDS clamp is unclamped, the total noise power becomes the sum of the noise contributions from the pixel follower amplifier and column bias transistor. The two noise sources present are 1/f and transistor shot

noise. The noise in the currents of these transistors creates a total noise current as follows:

$\langle i_{th}^2 \rangle$is the thermal noise contribution of each transistor:

$$\left\langle i_{th}^2 \right\rangle = \frac{8kTg_m}{3} \tag{2.17}$$

and $\left\langle i_f^2 \right\rangle$ is the contribution of the 1/f noise for each transistor

$$\left\langle i_{th}^2 \right\rangle = \frac{K_F I_b^{A_F}}{fC_{\text{ox}}L_{\text{eff}}^2} \tag{2.18}$$

the total noise current flowing in the column line is:

$$\left\langle i_{\text{col}}^2 \right\rangle = \left\langle i_{thM1}^2 \right\rangle + \left\langle i_{thM2}^2 \right\rangle + \left\langle i_{fM1}^2 \right\rangle + \left\langle i_{fM2}^2 \right\rangle \tag{2.19}$$

and the resulting noise voltage-squared energy to the input of the non-clamped CDS circuit is:

$$\left\langle v_{\text{col}}^2 \right\rangle = \frac{\langle i_{\text{col}}^2 \rangle}{g_{m1}^2} \tag{2.20}$$

The noise contribution of the follower amplifier is denoted by $\left\langle v_{\text{amp}}^2 \right\rangle$. For the APS buffer amplifiers it is calculated to be 185.54e-9V^2, corresponding to 431μV$_{rms}$[44].

In addition to all of these fundamental noise sources [45], there is also the unwanted variation in processing of the pixels that is collectively named "fixed pattern noise". The dominant phenomenon of fixed pattern noise is the random variation in the threshold voltage of the reset and pixel amplifier transistors. The Correlated Double Sampling circuit should eliminate the effects of this variation.

In addition, CDS should also eliminate the 1/f noise sources in the circuit as well. Because reset is sampled after the signal, and the two voltages are not part of the same integration cycle, the kT/C noise from the sampled reset is not correlated to the noise from the integration cycle. This means it is not eliminated, and so must still be included in the total noise figure. In fact, it must be counted twice, because the reset noise powers from the actual integration cycle and the sampled reset will add to each other. The noise contributions of the column *with* CDS cleanup of the 1/f noise as will be labeled as follows:

$$\left\langle v_{\text{colcds}}^2 \right\rangle = \frac{kT}{C_{\text{column}}} \tag{2.21}$$

Since the remaining noise after CDS only contains thermal components, it can be reduced to a kT/C term. This leads to a final noise expression, after CDS, of:

$$\left\langle v_{\text{apstotal}}^2 \right\rangle = \left\langle v_{\text{photon}}^2 \right\rangle + \left\langle v_{\text{dark}}^2 \right\rangle + 2\left\langle v_{\text{pixreset2}}^2 \right\rangle + \left\langle v_{\text{cdsclamp}}^2 \right\rangle + \left\langle v_{\text{colcds}}^2 \right\rangle + \left\langle v_{\text{amp}}^2 \right\rangle \tag{2.22}$$

or

$$\left\langle v_{\text{apstotal}}^2 \right\rangle = \frac{\left(I_{\text{photo}} + I_{\text{dark}}\right)\Delta t_{\text{reset}}}{C_{\text{pdiode}}^2} q + \frac{kT}{C_{\text{pdiode}}} + \frac{kT}{C_{CDS}} + \frac{kT}{C_{\text{column}}} + \left\langle v_{\text{amp}}^2 \right\rangle \tag{2.23}$$

Adding all of these noise contributions together gives:

$$\left\langle v_{\text{apstotal}}^2 \right\rangle = 304.3\text{e} - 9\text{V}^2 \tag{2.24}$$

for a final noise amplitude of 552μV_{rms}, neglecting 1/f noise.

2.3.1.3 APS Dynamic Range and SNR

The output voltage range of the APS system is originally limited by the maximum reset voltage in the pixel, minus the lowest voltage for reliable photodiode operation. For this imager, reset is approximately $V_{dd} - V_{TN}$, or 3.3V-1.0V=2.3V for an NMOS transistor including bulk effect. The reset transistor will still be supplying current to the photodiode during the reset cycle. Exactly how much current the photodiode draws is determined by the light intensity falling on the pixel at the time of reset, and the final voltage the pixel will reflect this. Since these factors can and do vary during the operation of the DPCT imager, the real reset voltage also varies. Part of the function of the Correlated Double Sampling circuit is to compensate for this normal variance of the reset signal. The follower in the pixel causes the column voltage to drop by another V_{TN}, which with the bulk effect reduces the maximum (reset) voltage on the column to 1.3V.

The minimum voltage of the column is dictated by the column current sources. They must stay in saturation. For this to be true, the column voltage cannot drop to less than $V_{ds} > V_g - V_T$, or

$$V_{ds} \geq \sqrt{\frac{I_D 2}{K'_N} \frac{L}{W}} \tag{2.25}$$

I_D is the saturated bias current for the column. For this chip and a bias of 260nA that is used in the columns, V_{ds} is calculated to be greater 85mV to stay in saturation. This gives a practical minimum voltage of 100mV. The output range is

therefore about 1.2V.

With an output range of 1.2V, and a noise level of 455μV_{rms}, the signal to noise ratio comes to 68dB.

2.3.1.4 APS Speed

The design goal for imaging frame rate was 30fps. The APS subsystem easily meets this specification for speed. Faster frame rates are possible, but there is a direct trade-off between exposure time and frame rate, faster rates necessitating higher light levels. The absolute limit on speed is limited by the column current sources that bias the source-followers in the pixels for operation. These current sources are normally biased to around 260nA for low-power operation. This current drive, combined with the column line capacitance of 200fF gives a maximum fall time of 1.3V/μs. This makes the worst case settling time for one column about 925ns with a 1.2V voltage range. The settling time for each CDS amp to be switched onto the pixel bus is 20ns. Thus, the columns in a row take 925ns to settle, and each pixel clocked out takes 20ns to settle. In the DPCT imager with 36 rows and 120 columns, the maximum frame rate is estimated to be 8300fps, ignoring the exposure problems associated with short integration time.

The minimum frame rate in practice is set by the desired SNR of the imager and the light level to be imaged. The maximum integration time per frame is 35/36 of

the frame duration. This gives the formula for maximum frame rate as:

$$r_{\text{frame}} \leq \frac{(\text{lightlevel})\,(\text{lighttovoltagegain})}{(\text{SNR})(\text{systemnoise})} \tag{2.26}$$

With values already computed this becomes:

$$r_{\text{frame}} \leq \frac{(\text{lightlevel})\left(244\frac{\mu\text{V}}{(\mu\text{W/cm}^2)\cdot\text{ms}}\right)}{(\text{SNR})(455\mu\text{V})} \tag{2.27}$$

2.3.1.5 Correlated Double Sampling Analysis

The CDS circuitry is essentially very simple: a capacitor, a switch, and a buffer opamp. Leakage of the floating capacitor node is the biggest potential problem to be faced, and here the severity of leakage on the output signal will be estimated.

From the dark current of the APS pixel, it is estimated that leakage from the drain diffusion of the clamping switch is 20.25aA. The CDS series capacitor has a value of 100fF, which leads to a voltage decay rate due to leakage of

$$\frac{\Delta V}{\Delta t} = \frac{20.25\text{aA}}{100\text{fF}} = 203\text{pV}/\mu s \tag{2.28}$$

For a 33ms frame, the 925μs row readout time will cause this voltage will decay by 188nV. This figure is far below the noise level of the APS system and can safely be ignored. Only a frame rate roughly 200 times slower than this (about 1 frame every 6.6s) would cause this leakage to be significant compared to the noise of the system.

Circuit	Current Consumption
Column Biases	31.2 μA
Photocurrent (APS)	8.64μA
Column (CDS) Buffer Amplifiers	417 μA
Digital Row Circuitry	16.4 μA
Digital Column Readout Circuitry	20.0 μA

Table 2.1: Estimated current consumption of APS circuits.

2.3.1.6 APS Power Consumption

Power consumption of the whole subsystem is the sum of digital row circuitry, pixel reset current, pixel amplifier output current, CDS circuitry, and finally the digital shift registers for outputting each pixel. Assuming a normal photocurrent of 2pA/pixel, which is observed under normal indoor lighting conditions, the total current of the APS subsystem is estimated to be 493μA. From Table 2.1, it is seen that the dominant current draw is the due to the biases of the CDS buffer amplifiers.

2.3.2 Centroid-Tracking System Analysis

Again, the subsystem analysis will be divided into two parts: pixel and edge circuitry. For the centroid-tracking pixel, sensitivity to light change and noise will

be analyzed. In addition, the linearity of the output circuit computation and general system characteristics will be examined.

Figure 2.5 shows a schematic of the centroid-tracking pixel, and Figure 2.6 shows one cell of the edge circuitry.

2.3.2.1 Centroid Pixel Sensitivity

It was described in section 2.2.2 that the equation for input voltage change necessary to trigger the centroid-calculating pixel was

$$\Delta V_{\text{in}} = -\frac{\Delta V_{\text{out}} U_t}{\kappa V_{0P}} \tag{2.29}$$

or

$$\Delta V_{\text{in}} = -\frac{(V_{TN} - V_{\text{invreset}})\, U_t}{\kappa V_{0P}} \tag{2.30}$$

In addition to the ΔV_{in} necessary to raise the output of the inverter from its reset state to V_{TN}, there is also the coupling of the reset switch and dummy compensation switch to consider. They will add a ΔV_{switch} voltage to the input that will significantly affect ΔV_{in}. While the reset switch is being turned off, it still has some finite resistance. In addition, the output of the inverter remains a low-impedance restoring voltage, which can sink enough current to offset the charge from the gate of the switch as it turns off. Therefore, most of the effect of the clock feed-through will occur as the NMOS switch gate voltage goes below V_{TN}. In this region of weak inversion, most of the charge has already been depleted from the

channel. The remaining clock feed-through effect resides in the gate-drain overlap capacitance. This capacitance for a minimum-size transistor is about 0.5fF. Including bulk effects, the gate voltage at which the switch will turn off will be around 1.6V. The charge injected by a voltage swing of 1.6V into 0.5fF is only about 0.8fC. This amount of charge can be removed by the tiny subthreshold currents running through the switch before it is completely off. These currents can easily be on the order of 1nA, even at very low gate voltages. 1nA would remove 0.8fC in 0.8μs. As long as the fall time of the clock signal is even slightly slow, the effect of clock feed-through will be reduced by these subthreshold currents. The charge that does feed through will arrive at the main 150fF capacitor of the pixel. In addition, the dummy switch transistor operating in full inversion will take in more charge with its complementary clock than the main switch manages to release. It will couple into the input node with both drain and source overlap capacitances, and with the gate-channel capacitance while above-threshold. The combined effect of both transistors, conservatively assuming the main switch does not conduct very well while it is in subthreshold, is as follows:

$$\Delta V_{\text{switch}} = -V_{TN}\left(\frac{C_{gd1}}{C_{\text{accap}}}\right) + \left[V_{dd}\left(\frac{C_{gd2}+C_{gs2}}{C_{\text{accap}}}\right) + (V_{dd}-V_{TN})\left(\frac{C_{gc2}}{C_{\text{accap}}}\right)\right] \quad (2.31)$$

It should be noted that the "high" gate voltage of the dummy switch need not be V_{dd} as it is in this design. By changing the value of the logic high voltage sent to the gate of the dummy transistor, the expression for ΔV_{switch} would be different. The value of this variable voltage would be substituted wherever V_{dd} appears in

the current formula. In this way, one could directly control the value of ΔV_{switch} by this control voltage. This would in turn control the sensitivity of the pixel.

The completed expression for ΔV_{switch} allows us to write the full description of the change in photodiode voltage necessary to trip the inverter.

$$\Delta V_{\text{pdiode}} = \Delta V_{\text{in}} + \Delta V_{\text{switch}} \tag{2.32}$$

Normally, biases and voltages are set such that A_{inv}=-1260 and ΔV_{out}=250mV. ΔV_{in} thus becomes 198μV. ΔV_{switch} is computed to be 56.5mV. The voltage for ΔV_{pdiode} in this case is dominated by ΔV_{switch}.

The photodiode voltage is regulated by V_{gs} of transistor M1. This V_{gs} is directly dependent on the photocurrent of the photodiode. If the light falling on this pixel induces a subthreshold pixel current, (as it does for almost all lighting conditions,) then the source voltage of M1 will change as the natural log of current change. The current will have to increase by a specific multiplicative factor from its value during inverter reset to change V_{gs} by a sufficient amount. In order to change the source voltage by a specific ΔV_{pdiode}, the current will need to reach I_{trip} as described in the following equation:

$$I_{\text{trip}} = M_{\text{light}} I_{\text{reset}} \tag{2.33}$$

$$M_{\text{light}} = \exp\left(\frac{\Delta V_{\text{pdiode}}}{U_t}\right) \tag{2.34}$$

From the values for ΔV_{pdiode} given above, M_{light} = 9.6.

2.3.2.2 Centroid Pixel Noise

Since the output of each centroid-tracking pixel is a digital voltage, noise of the output voltage is not a concern. Noise can degrade the operation of the pixel by introducing a random voltage component into ΔV_{in}, and hence in the factor increase in light level necessary to activate the pixel. In this section this noise-induced uncertainty will be calculated.

The first noise source arises from the process of resetting the inverter. During reset, the output node of the inverter will exhibit kT/C noise due to the thermal noise in the inverter transistors, and the main pixel capacitor. The main pixel capacitor is 150fF, so this reset noise is: $\langle v_{\mathrm{reset}}^2 \rangle = 26.5\mathrm{e}-9\mathrm{V}^2$, for an amplitude of 162$\mu\mathrm{V}_{rms}$. When reset is released, the input will start with this level of noise.

The photodiode/NMOS bias sub-circuit is easy to analyze for noise, because it is the same as an APS photodiode in perpetual reset. As such, it also exhibits kT/C noise. Because during normal (non-reset) operation the main explicit capacitor in the pixel is floating, it will not contribute to this noise figure. The photodiode parasitic capacitance on this node totals 82fF. This gives $\left\langle v_{\mathrm{pdiode}}^2 \right\rangle = 50.5\mathrm{e}-9\mathrm{V}^2$ and a noise amplitude of 225$\mu\mathrm{V}_{rms}$.

These two noise sources together give an noise power to the input of the inverter of

$$\left\langle v_{\mathrm{reset}}^2 \right\rangle + \left\langle v_{\mathrm{pdiode}}^2 \right\rangle = 77\mathrm{e}-9\mathrm{V}^2 \tag{2.35}$$

or an RMS noise voltage of 277.5μV. That much voltage change on the input of

the inverter corresponds to an extra factor of 1.01 of photocurrent and hence light intensity. Compared to the threshold for tripping the inverter of M_{light} = 9.58, this is a small amount of noise—clearly not enough to cause accidental activation of the inverter.

2.3.2.3 Centroid Subsystem speed

The limiting speed of centroid operation is dependent on the reset time and the propagation delay of the pixel inverter. There is a certain minimum time that the inverters of the pixels need to be reset to settle to their final trip point.

Each inverter during reset can be modeled as a PMOS current source and a diode-connected NMOS transistor, along with the main pixel capacitor to AC ground. The diode-connected NMOS can be approximated as a resistor with value $1/g_{m5}$, with g_{m5} calculated at the operating point where V_{out} has reached its final equilibrium voltage. This will not be a true representation of the circuit operation. An accurate analysis requires a large signal model. But the value for $1/g_{m5}$ at the equilibrium point will give a result that is only more conservative than the true behavior. When V_{out} starts too high and must fall to reach equilibrium, the actual g_{m5} will be more than the equilibrium g_{m5}, and the circuit will reach equilibrium faster in reality. When V_{out} starts lower than equilibrium, the actual g_{m5} will be lower, and the circuit will again reach equilibrium in reality faster than calculated in theory.

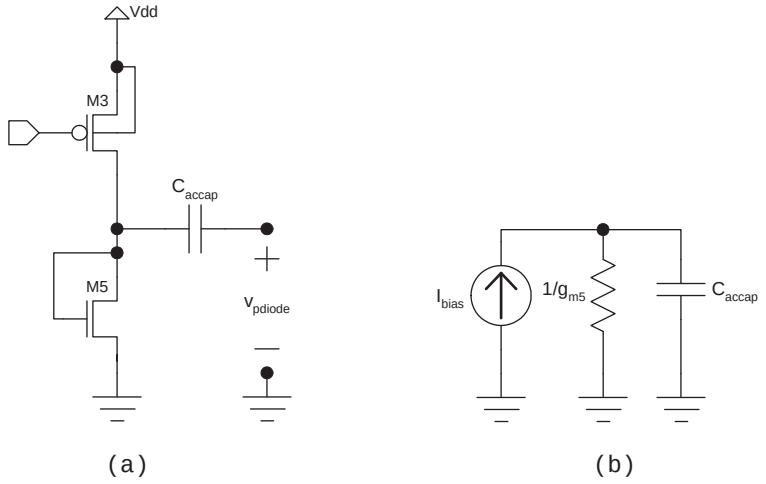


Figure 2.7: (a) Centroid pixel inverter in reset. (b) Equivalent small-signal inverter reset schematic.

Figure 2.7(a) shows the simplified inverter circuit and Figure 2.7(b) its equivalent small-signal circuit. In this simplified model, it becomes a simple RC circuit, with $\tau = C_{\rm accap}/g_{m5}$. $g_{m5} = I_D(\kappa_N/U_t)$ for this circuit in subthreshold. At equilibrium, I_D=68.82pA, so g_{m5}=2.27e-9mho and R=439.8MΩ. With $C_{\rm accap}$=150fF, this gives a time constant of τ=66.0μs. (2.2)τ, or 145μs, is a more conservative time constant, with the voltage moving 90% of its total swing in that time. The actual reset time constant will be shorter, but this is a good conservative minimum.

The propagation delay of the inverter is directly dependent on the bias of M3. The parasitic capacitance on the output of the inverter is roughly 12.3fF. This yields a propagation delay of

$$t_{\rm inverter} = \frac{Cv_{TN}}{I_{\rm bias}} = \frac{(12.3{\rm fF})(0.75{\rm V})}{I_{\rm bias}} \tag{2.36}$$

For 68.82pA, $t_{\rm inverter} = 134\mu$s. Summing this with the minimum reset time gives

a total minimum cycle time of 279μs and a maximum centroid rate of 3580Hz. It should be noted that increasing the inverter bias current by an order of magnitude will decrease the inverter propagation time by a factor of 10, and will increase the centroid system current consumption by only about 1.5%.

The time after reset and before the tripping of the inverter is spent waiting for the photocurrent to change by a sufficient magnitude. The length of this period of time, t_{detect} determines the minimum rate of change of light levels in order to be detected.

$$R_{\text{change}} = \frac{M_{\text{light}}}{t_{\text{detect}}} \rightarrow t_{\text{detect}} = \frac{M_{\text{light}}}{R_{\text{change}}} \tag{2.37}$$

For a desired maximum R_{change}, equation 2.37 computes the minimum t_{detect} part of the centroid cycle. The final cycle time takes

$$t_{\text{cycle}} = t_{\text{reset}} + t_{\text{detect}} + t_{\text{inverter}} \tag{2.38}$$

Conversely, the longest wait without the inverter falsely tripping determines the maximum period possible with the centroid circuit. Leakage from the switch drain and source diffusions limits the amount of time the input of the inverter can remain floating. Taking ΔV_{in}=198μV from section 2.3.2.1, the leakage current of 60.75aA from the three drain/source diffusions, and 150fF capacitance of the input to the inverter, the maximum time before the leakage current alone causes the inverter to trip can be predicted. In the absence of light, this should happen at Δt_{leak}=489ms.

Circuit	Current Consumption
Photocurrent (6pA/pixel)	12.96nA
Digital Circuitry (180Hz)	305 pA
Pixel Inverters (68.8pA/pixel)	148.7 nA
Resistive Ladder Diffamps (1.2μA/amp)	115.2 μA
Resistive Ladder	4 μA

Table 2.2: Estimated current consumption of centroid-calculating circuits.

2.3.2.4 Centroid Power Consumption

The power consumption of the centroid-tracking circuitry depends on the photocurrent drawn by the continuously biased photodiode, the operation of the pixel inverter, and the digital and analog circuitry on the periphery of the chip.

Photocurrent can easily vary by decades depending on the intensity of the incident light. We will assume a photocurrent of 6 picoamps, which has actually been observed in indoor lighting conditions with the DPCT chip. Given this level of incident light, the continuously-biased pixels use about 13nA over the whole array. The total current of this block has been estimated to be 116μA, of which the largest component goes to the buffer diffamps on the resistive ladder of the edge circuitry. Table 2.2 shows the computed current draw of each part of the centroid circuitry.

Chapter 3

Imaging and Motion Centroid Tracking Array: Results

3.1 APS Performance

3.1.1 Measured APS Linearity, Gain, Sensitivity

(See Figure 3.1 for example APS images.) A graph of pixel voltage output vs. light intensity input is shown in Figure 3.2. Voltages corresponding to the input light range from $11\mu W/cm^2$ to $121\mu W/cm^2$ were fit to the straight line in the graph. Within this range, the slope was $8.27mV/(\mu W/cm^2)$. For the chosen full-rate video integration time of 33ms, this corresponds to $250\mu V/((\mu W/cm^2)\cdot ms)$, only 3.3% off from my $242\mu V/((\mu W/cm^2)\cdot ms)$ estimate in the Analysis section.

The voltage range corresponding to this fit was 2.19V to 3.09V, for a linear range

Figure 3.1: Example pictures from APS image array.

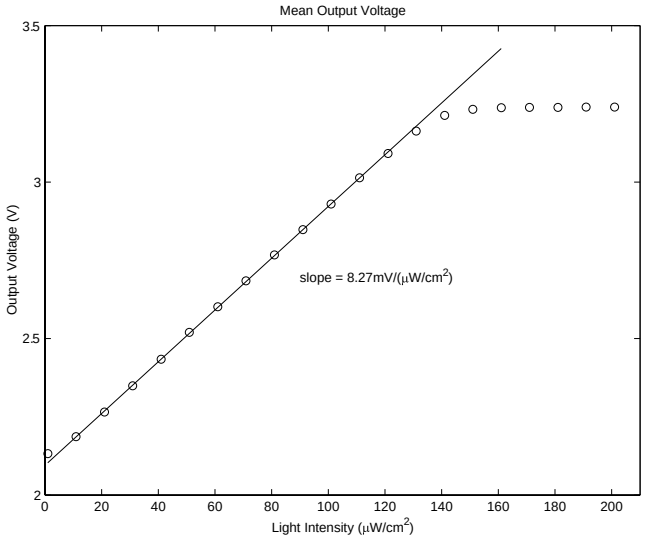


Figure 3.2: Output pixel voltage versus input light level, 33 ms integration time.

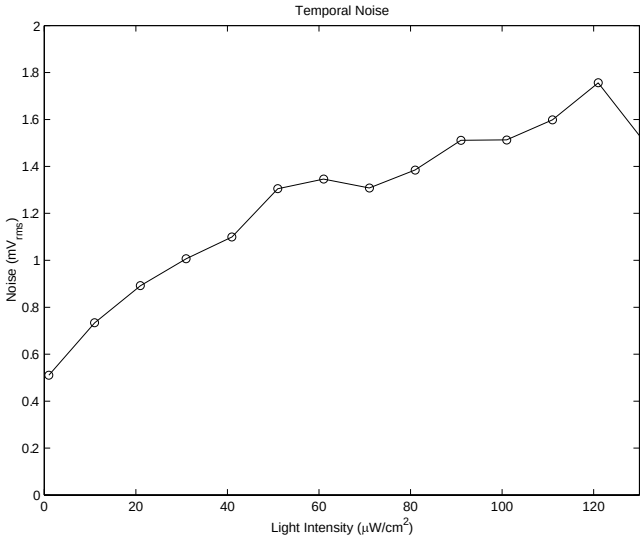


Figure 3.3: Temporal noise at output of chip.

of 0.9V. Within this range, the RMS error of the voltage values from the fit line was 2.3mV.

3.1.2 Measured APS Noise

(Figures 3.3, 3.4, and 3.5 show graphs of measured APS noise.) All measured noise of the APS imager exceeded predictions from the analysis. This is understandable, considering that it was necessary to omit 1/f noise from the computed noise value.

Temporal noise in the APS imager, measured over time at the same pixel location, was extremely low compared to other noise sources. All measurements indicate a temporal noise of less than 1.8mV$_{rms}$ over the linear signal range. See

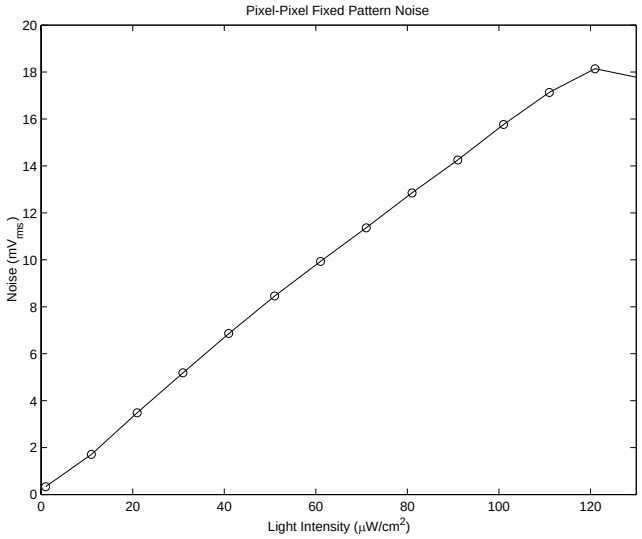


Figure 3.4: Pixel-Pixel Fixed Pattern Noise.

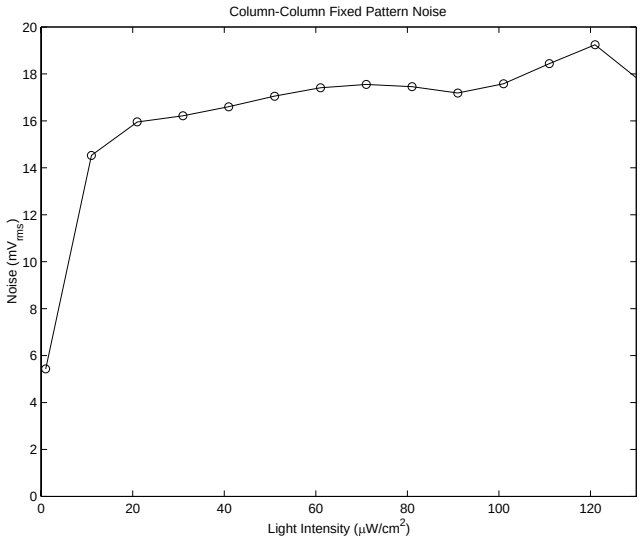


Figure 3.5: Column-Column Fixed Pattern Noise, entire image.

Figure 3.3. The predicted APS system noise was 552μV_{rms}, which corresponds to the minimum value of the temporal noise vs. light level graph. The noise of this measurement was so low, that it is probably approaching the minimum resolvable levels that the board readout electronics are capable of accurately detecting.

Pixel-to-Pixel fixed pattern noise for the imager was barely detectable at low light levels, increasing at an almost linear rate as the signal level increased. Looking at the large-signal equation for a gate-source follower with transistor M1 the follower and M2 the current source,

$$V_{\text{in}} = V_{\text{out}} + V_{T0} + \gamma_1 \sqrt{2\left|\phi_F\right| + V_{\text{out}}} - \sqrt{2\left|\phi_F\right|} + \sqrt{\frac{I_{\text{bias}}}{K'_N \left(\frac{W}{L}\right)_1}} \sqrt{\frac{1 + \lambda_2 V_{\text{out}}}{1 + \lambda_1 \left(V_{dd} - V_{\text{out}}\right)}} \quad (3.1)$$

we can then derive an expression for the change in output voltage due to a change in input voltage (as would be computed by the CDS unit)

$$\begin{aligned} V_{\text{in}}(\text{reset}) - V_{\text{in}}(\text{final}) &= V_{\text{out}}(\text{reset}) - V_{\text{out}}(\text{final}) - \gamma_1 B - \sqrt{\frac{I_{\text{bias1}}}{K'_N \left(\frac{W}{L}\right)_1}} D \\ B &= \sqrt{2\left|\phi_F\right| + V_{\text{out}}(\text{reset})} - \sqrt{2\left|\phi_F\right| + V_{\text{out}}(\text{final})} \\ D &= \sqrt{\frac{1+\lambda_2 V_{\text{out}}(\text{reset})}{1+\lambda_1 (V_{dd} - V_{\text{out}}(\text{reset}))}} - \sqrt{\frac{1+\lambda_2 V_{\text{out}}(\text{final})}{1+\lambda_1 (V_{dd} - V_{\text{out}}(\text{final}))}} \end{aligned} \quad (3.2)$$

From equation 3.2, we can see that the deviation from a direct $\Delta V_{\text{in}} = \Delta V_{\text{out}}$ relationship involves the bulk effect of the pixel amplifier transistor (B) and the drain conductance of both pixel and column current source transistors (D). It can be shown that the factor D increases almost linearly with decreasing V_{out} given λ_1=0.0801, and λ_2=0.0626, which is the case for the APS system. As D increases, it magnifies the contribution of the $(W/L)_1$ factor from the gate-source follower in

the expression. Thus, any variation in the gate-source follower transistor geometry in the pixel will have an increasing effect as the signal increases and V_{out} decreases. To decrease this effect, we need to reduce λ_1 and λ_2. Lengthening the column current source transistor, and also the pixel transistors if space allows, will accomplish this.

Column-to-column fixed pattern noise results were relatively constant over the linear range of the imager. They stayed at or near the worst pixel-to-pixel noise of about $16\mathrm{mV}_{rms}$. There are two chief reasons for this undesirable performance. First of all, because the simple architecture of the DPCT used a CDS circuit on the end of every column and not globally, there is no way to correct column-to-column offsets. Use of a global CDS circuit would alleviate much of the column-to-column FPN.

The second main reason has to do with the layout of the pixel array. Alternating lighter and darker vertical stripes are apparent in images of dimly, evenly lit scenes. Because two APS pixels are tiled above and below every one centroid-mapping pixel in the array, the physical and electrical environment of adjacent APS pixels is not the same. Put more simply, for every two adjacent APS pixels, one will see the left sides of neighboring centroid-mapping pixels, and the other will see the right sides of neighboring centroid pixels. Because of the asymmetric placement of the photodiode area within the centroid pixel, doping profiles for left and right APS pixels will be slightly different. Also, due to the differing proximity

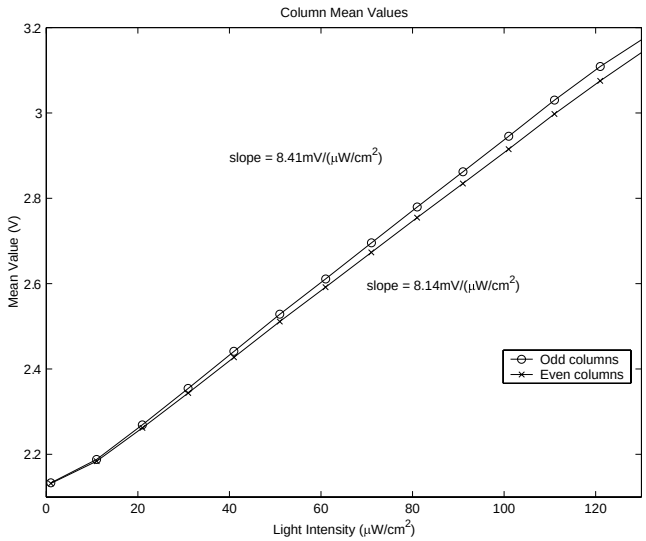


Figure 3.6: Voltage/Light transfer plot for odd and even columns.

of the centroid pixel's photodiode, the amount of photo-generated carriers in the substrate will be different for left and right APS pixels for the same incident light. These are unfortunately types of gain errors, and as such is not able to be remedied by the CDS circuit. This phenomenon is apparent when measurements are taken separately for odd and even column groups. Figure 3.6 clearly shows a different light-voltage transfer function for odd columns than for even columns. Looking at the average pixel value of every column of the array in an evenly-lit scene, it becomes apparent that the distribution is not normal. In fact, the entire array experiences a bi-nodal distribution, due to the two separate and distinct mean values observable especially at higher light levels. The column-to-column FPN of all odd columns and the FPN of all even columns taken separately are each better than the

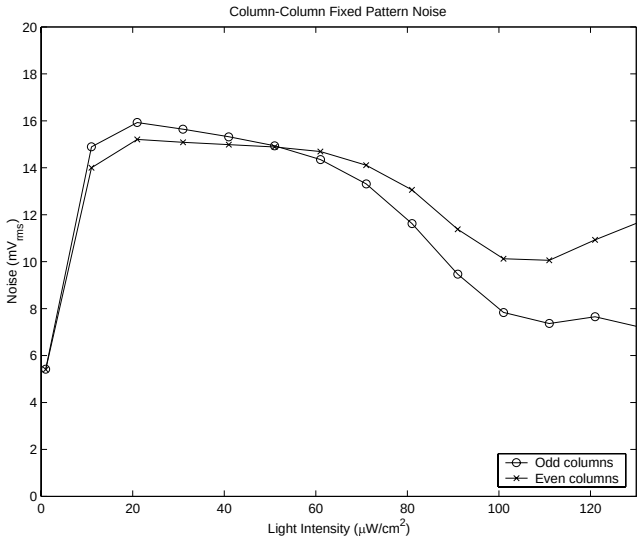


Figure 3.7: Column-Column fixed pattern noise for odd and even columns.

combined FPN figure. Figure 3.7 shows even and odd statistics separately.

In future chips, barrier wells could be used to isolate APS pixels from centroid pixels and improve performance, at the expense of area and fill factor of the array. Another solution would be to make the array perfectly regular, with one APS pixel for every centroid-computing pixel, or to make the centroid-tracking pixels perfectly symmetric.

3.1.3 Measured APS Dynamic Range and SNR

The total noise of the APS imager as a percentage of signal level for different light levels can be seen in Figure 3.8. At maximum signal level, the total noise (standard deviation/mean signal level) of 2.88% corresponds to an SNR of 30.8dB.

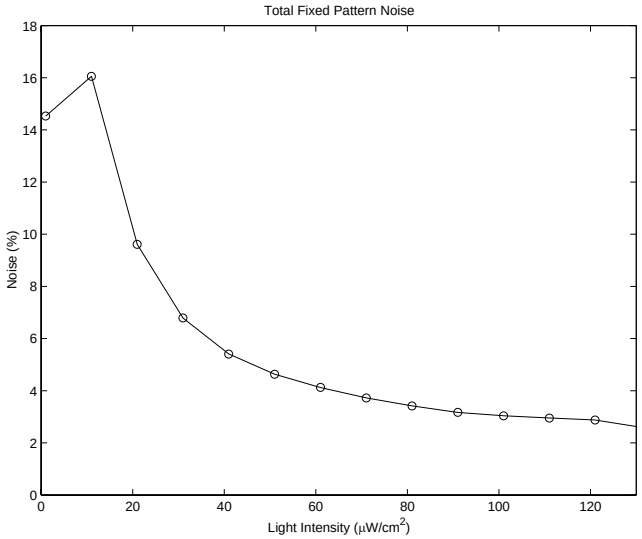


Figure 3.8: Total FPN noise percentage for different light levels. (standard deviation/mean signal level)

3.1.4 Measured APS speed

Imaging at a high enough frame rate to test the limits of the APS is difficult, due to the high light levels necessary for a useful signal. The readout circuitry was tested for speed, however. The readout circuits still functioned properly up to a pixel clock speed of 11MHz, a 91ns period. This was also the limit of the test circuitry.

3.2 Centroid Computation Performance

3.2.1 Centroid Frequency Response

From section 2.3.2.3, it was estimated that the fastest rate of blinking for a stationary blinking light to remain detectable was 3.6kHz. To test this calculation, a blinking LED was used as a visual target, fed with a square-wave signal of sufficient amplitude and variable frequency. The frequency at which it ceased to be detectable was around 4.2kHz.

At the end of the same section, the slowest reset possible was calculated to be 489ms. To confirm this, the array was covered to protect it from light, and let sit until the pixels tripped through leakage. The actual measured time varied between 450ms and 490ms in the dark, and took about 200ms in ambient light with no motion. This strongly suggests that the drain/source diffusions of the inverter switches are leaking by either receiving indirect light, or are being affected by minority carriers in the substrate from the nearby photodiode. Such effects could be reduced by more careful layout of the switch transistors.

3.2.2 Centroid System Performance

The centroid tracking system was tested using an analog oscilloscope screen as a target. The X/Y mode setting was used, and two function generators set to 10 and 20 Hz supplied the scope channels. In this way, a moving point of light tracing a

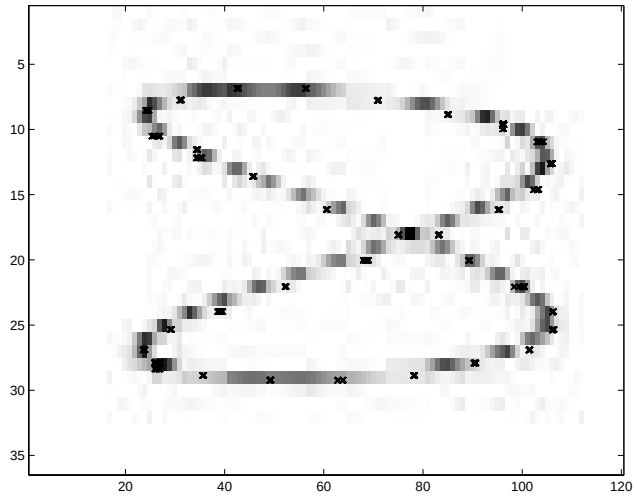


Figure 3.9: Position plot of output centroid data from chip and sum of a reverse-video series of APS images in background.

stable "figure-8" pattern could be observed on the oscilloscope screen. APS image data and centroid coordinate data were taken simultaneously. Centroid voltages were converted to digital data, and sent to a controlling computer. A composite image of all APS frames was produced by summing all frames and then inverting the brightness of the image for easier printing. On top of this composite image is plotted the centroid positions reported by the centroid-tracking subsystem of the chip. The result is displayed in Figure 3.9. The data is an excellent match of the target, which was comprised of two sine waves in the x- and y-directions. There are six centroid coordinates taken for every APS frame taken. One such APS image and centroid coordinates of the current and previous frame are displayed in Figure 3.10. It is obvious that while the APS imager sees one smear of the path of the oscilloscope point, the centroid-tracking circuitry is able to accurately and

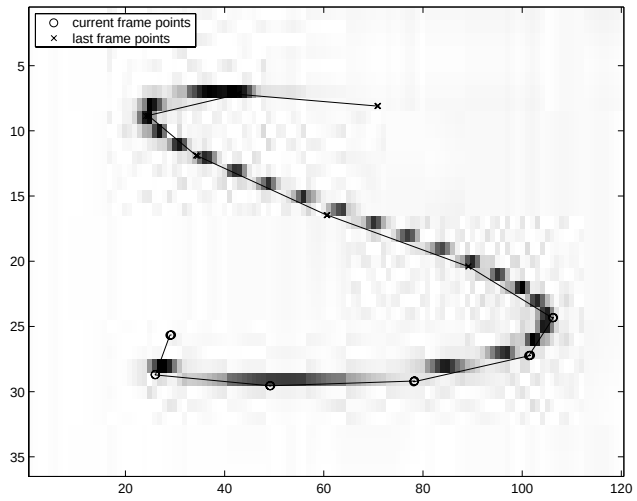


Figure 3.10: Reverse-video of one APS frame image with 6 corresponding centroid positions, and 6 positions from previous APS frame.

precisely plot specific points along the path in real time.

Figure 3.11 shows another example of the cumulative centroid positions reported for an oscilloscope target. This time, a non-blinking, stationary LED was placed next to the moving oscilloscope target to show that the stationary LED has no effect on centroid positions despite the fact that it is much brighter than the oscilloscope.

With faster moving targets, the speed of the centroid subsystem could be increased even more. Centroid pixels are sensitive to changes in incident light since their last reset. Therefore, faster changes in light (faster movement) would allow for shorter reset intervals and higher measurement frequency.

In addition to trials involving a single moving target, experiments using the chip with multiple targets were performed. In the first, a target consisting of 3

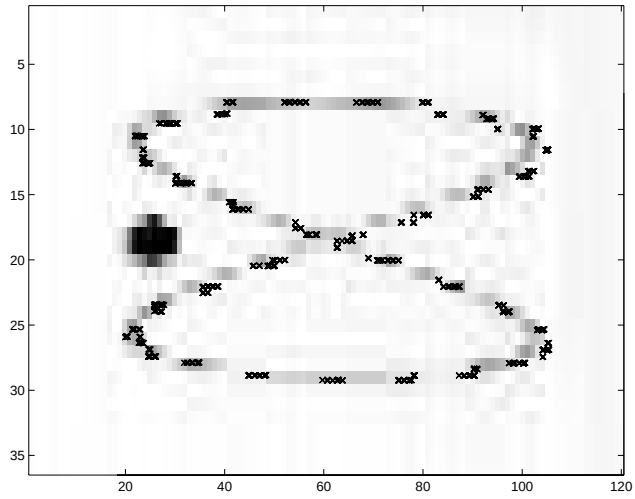


Figure 3.11: Reverse-video sum of APS images of figure-8 and stationary LED, with all centroid positions.

LED's laid out in a triangle formation was imaged. All LED's were either blinking or steadily on, and were stationary. Three different tests were performed. The first test involved all three LED's blinking at exactly the same time. Figure 3.12(a) shows a histogram of the centroid positions reported by the chip, with blinking LED positions marked by circles. From this histogram, we can see that the vast majority of positions reported are in the center of the 3 LED positions. Notice that since two LED positions are on nearly the same row, their contributions to the row position of the centroid are overlapping. Due to the previously mentioned method for centroid determination which ignores the number of active pixels in a row, the computed centroid can be seen closer to the far point of the LED triangle than would be expected from a true centroid. The weight of the far point (row 23) in the centroid computation will be comparable to both LED's together on row 10

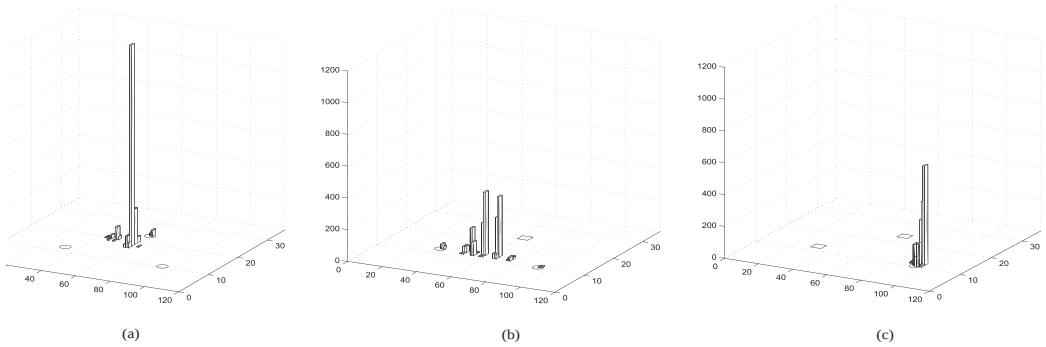


Figure 3.12: 2-D Histograms of centroid response with a target of three LED's. Circles indicate blinking LED positions, squares indicate steady-on LED positions. (a) 3 blinking (b) 2 blinking, 1 steady-on (c) 1 blinking, 2 steady-on

of the graph. The second experiment was the same as the first, except that one LED was made to be continuously on and not blinking. In Figure 3.12(b), the non-blinking LED location is marked with a square outline instead of a circle outline. Again, the positions plotted lie in-between the two blinking LED positions, and are unaffected by the steadily-on LED. Similarly, Figure 3.12(c) shows a test with one blinking LED position marked with a circular outline, and two non-blinking steadily-on LED's marked with square outlines. In this case there is no doubt that the only positions reported are right on top of the only element of the scene that is changing in time.

Another experiment with multiple blinking LED's was performed that involved uncorrelated blinking. Two LED's with separate blinking periods and phases, at different x- and y-positions, were set up in front of the DPCT imager and centroid positions were recorded. Figure 3.13 shows a histogram of the number of values

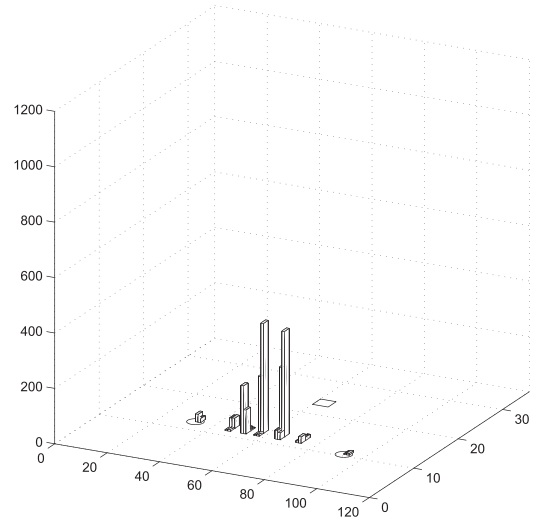


Figure 3.13: 2-D histogram of imager array, showing reported centroid positions for two uncorrelated blinking LED's

recorded at a specific region of the array. It can be seen that in addition to the two positions of the actual LED's showing a marked response, the linear combination of their positions also shows a considerable number of recorded coordinates. If two LED's are seen to blink in the same period of time t_{detect}, the centroid of their positions will be computed and reported. This is the normal operation of the centroid subsystem. Multiple target tracking is still possible, however, with the addition of some basic statistical analysis of the positions reported. Through techniques such as SVD, the linearly-independent positions can be extracted, and the linear combination of the two positions can be recognized as a false position. Of course this has more limited applicability. For instance, if true movement of a third object happened to coincide with the linear combination of the movement of other two objects, it might be falsely omitted. But for simple observations of a

few objects, it is possible to extract meaningful position data all objects involved. A system with broader applicability could be constructed by changing the edge circuitry of the centroid subsystem, allowing the location of multiple regions of activity in the array. This is a hardware change to be pursued in the next-generation DPCT imager.

3.3 Summary

An imager chip has been built and tested having an array composed of two types of pixels. One type, an Active Pixel Sensor, is optimized for low-noise imaging, while the other type is a custom designed pixel for computation of the centroid of moving objects in the image. The APS array contains 36 x 120 pixels, and the centroid-computing array contains 36 x 60 pixels. Having two types of pixels allow each to be optimized for its separate task. One consequence of the mixed nature of the array is that APS image quality has suffered. This is not an intrinsic problem, and can be solved with modified layout of the pixels. The centroid-tracking system performed well, successfully locating the position of a moving target in real time much faster than the APS imager could merely take images. It is by default a single-target tracker, though off-chip methods are available to expand its abilities to track multiple targets.

Technology	0.7μm 3M1P CMOS
Array Size	APS: 120 (H) $\times$ 36 (V) Centroid: 60 (H) $\times$ 36 (V) Total Size: 1.6mm $\times$ 1.8mm
Pixel Size	APS: $14.7\mu\text{m} \times 14.7\mu\text{m}$ Centroid: $29.4\mu\text{m} \times 29.4\mu\text{m}$
Fill Factor	APS: 16% Centroid: 12%
Power Consumption **(3.3V Supply)**	2.6mW
FPN (APS) **(Std. Dev./Full Scale)**	2.88%
Dark Response (APS output)	6.8mV/s
Conversion Gain (APS)	$1.7\mu\text{V/e}^-$
Output Voltage Range (APS)	0.9V

Table 3.1: Summary of chip properties.

Chapter 4

Imaging and Motion Centroid Tracking Array: Applications

4.1 Introduction

With the functionality of the combination DPCT imager established, we now turn to practical uses of this technology. The essence of the imager is the central theme of this thesis: local processing at the sensor in order to generate and communicate only useful or interesting data. "Interesting" in this case is defined as that part of the scene which is changing. Temporal change of the scene is not the only possible metric, of course, but it is a useful criterion for many situations, for the simple reason that dynamic elements represent new information. In the rest of this chapter, we will explore some example applications where noticing and tracking

changes in the scene is useful.

4.2 Mobile Robotics

Most robots used today are not shiny anthropomorphic servants such as C-3P0 from the movie *Star Wars*. Typically they are factory robots, optimized for a single type of task in a carefully controlled setting. On an assembly line, a robot which consists of a single arm or multiple arms can take care of repetitive tasks efficiently.

Of course, the dream of people and roboticists alike is to enable more autonomously operating robots. To be even primitively autonomous, a robot should be able to be mobile, or roam about. Mobility in itself is not a problem, but navigating indoor or outdoor terrain is much more challenging. Navigation includes sensing the environment, planning movement, and responding to changes in that environment. In addition to navigation, to be more than a travelling observer a robot should be able to carry out useful motor tasks, such as manipulating objects.

It's not a mental stretch to see that visual information is useful for these tasks, and indeed many mobile robot designs utilize a camera. Typically they include a standard CCD-based imager with considerable processing using a PC or similarly-sized computer to process all of the images streaming from the computer. This approach works well for people because CCD-based cameras and PCs are ubiquitous, and so is the knowledge to use them. Unfortunately the combination is not

particularly energy efficient. To scale robots down to centimeter-size instead of meter-size, their batteries must scale down as well. To enable such smaller, nimbler robots with less energy storage, image sensing and processing must become more energy-efficient [46].

With the DPCT imager, a robot can do a few things important to its functioning. While its camera is stationary, it can notice and track moving objects around it. Any idea or map of the robots environment is sufficient as long as the environment is unchanging. Moving objects are likely to be the most deserving of notice, as they are the parts of the robots environment that are dynamic. It is essential to notice and keep track of changes to the environment if the robot is to continue to navigate it successfully.

Another important task is the manipulation of objects. This type of task is aided by the DPCT imager. The object-tracking functionality of the imager has a fast output rate—it can output centroid positions of a moving object easily at 180positions/s. In addition to having a fast system throughput, it also has extremely short latency. Since its computation is not pipelined, computation is near instantaneous, taking as long as it takes to report one position, or less than 1/180 seconds. This short latency is a clear advantage in any kind of motor feedback application, where corrections to robot movement are computed from the visual information from the camera. Low latency allows for quick movement without instability in the feedback loop. In a signal chain from a CCD camera, there is Analog-to-Digital

conversion, data transmission, and finally a separate pipeline for the actual computation. Thus even if position tracking occurs at the standard 30frames/s frame rate, the actual latency will be longer than 1/30 second for the change to progress through the whole processing chain.

All of this functionality is achieved with extremely low power consumption. The 2.6mW consumed by the DPCT imager to produce a low-noise APS image at 30f/s and compute centroids at 180pos./s is many orders of magnitude smaller than the power consumed by a CCD camera and PC hardware.

4.3 Biomedical Applications

4.3.1 Surgical Training and Evaluation

In micro-surgery, such as surgery on the retina or some vascular surgeries, actions requiring near-micron-precision are commonplace. While microscopes can be used to see a greatly magnified view of the operating area, the surgeon's hand still controls the operating instruments directly. These procedures thus demand enormous precision and steadiness. Mastering microsurgery demands both skill and training in order to minimize hand tremor.

To examine the effects various environmental factors on hand tremor, Tomlin [47] used the another motion-tracking chip to measure the position and movement of a tool in the hands of test subjects. The study explored the idea of using biofeed-

back to help surgeons to reduce their tremor, and examined the effects of exercise and fatigue on tremor. The data from these experiments yielded useful data that can be used to help surgeons increase their performance.

The same setup can be used to give direct feedback on tremor for surgeons in training. This will help beginning surgeons understand their own performance and how to improve it. In addition, it is a useful way to monitor surgeons and their performance in the field. Measuring the effects of fatigue on individual surgeons can give them knowledge of how their own performance is affected by fatigue. Aggregate data can also help shape policies directing the working hours of micro-surgeons in order to maintain surgical efficacy and safety.

4.3.2 Computer-Assisted Surgery and Tool Tracking

Another exciting application of optical-tracking technology is to actually help remove the tremor of the surgeon's hand. Actuators in the tool, if properly directed, could move the tool opposite to the surgeon's tremor and cancel out the unwanted movement.

The steady-hand robot [48] accomplishes this cancellation by sensing tremor from the force of the human operator's hand on the surgical tool itself. To work properly, this method depends on the surgical tool being fixed in space semi-rigidly, using a control arm mounted on a stationary platform. The large-scale movement of the tool is thus restricted, in addition to tremor being cancelled.

While effective, a certain amount of freedom of movement is given up for mechanical sensing of the tool to work correctly.

Optical tracking of the tool offers an alternative to mechanical sensing of position. For an optical system to be part of a feedback loop, there needs to be much more demanding requirements on both the update rate of the position sampling and latency, or delay. Because hand tremor occurs roughly in the range of 4–15Hz, a feedback loop would need to operate much faster than this to be both effective and stable. And even with speed, considerable delay between sensing and reporting of motion will also cause instability. Both of these requirements cannot be met with a traditional imager operating at 30fps, and computation system delays of visual input to position out typically greater than one frame time. However, the centroid-tracking imager can easily operate at 180positions/s, as described earlier. With no additional delay from sensing motion to reporting position, latency is always less than 1/180s. It is believed that this uniquely qualifies the DPCT imager for this important optical feedback task.

With an optically sensed instrument, there is no longer need for the instrument to be fixed in space. There is no longer a need for the instrument to oppose motion merely in order to sense movement. Because force sensing is no longer necessary, the surgeon can move the instrument more naturally. And micro-actuators on the tip of the tool should allow for the magnitude of hand tremor to be significantly reduced.

4.4 Distributed Sensing Applications

Low-power imaging systems are extremely important for isolated applications that can supply only limited power, or must rely completely on a finite supply of stored power. Such situations include duty in hazardous locations, such as nuclear reactors or military surveillance in enemy territory, and locations unavailable for maintenance, such as spacecraft. Low-power systems are also called for in any other location where economy or convenience rule out connection to dedicated power or constant replacement of batteries.

Constructing a low-power camera is clearly called for in such situations. But imaging is only part of the power budget for an isolated device. Without wires for communication, wireless communication is the only option for conveying images back to a remote operator. This can often be the largest use of power in a remote system. A survey of consumer wireless cameras reveals that transmission power of 200mW is common for indoor communication. For a remote imager farther away than a few hundred meters, the transmit power would need to be even greater. To ensure complete low-power system operation, communication must be kept to an essential minimum. Compression is one option, but comes with the cost of extra power used in computing compression algorithms. The simplest way to get significant power savings in many surveillance situations is to take advantage of the fact that often the scene being observed is static. By sending data only when movement in the scene is noted, large time spans can pass without using any transmit

power at all.

Conventionally, one would execute change detection with a conventional imager, ADC (often part of the imager chip), and microprocessor. All of these components would be running continuously, and if change in the scene were noticed, transmission of video data could be initiated. But even with the best components available, this strategy would still use far more power than the DPCT imager. To compare the two, we must compare the power used by the DPCT imager, and the conventional imaging signal chain as described above. We'll use 30fps as the rate at which we look for change, even though the performance of the DPCT imager is actually much faster.

From a survey of the most power-efficient imagers commercially available, "low-power" seems to indicate a power usage between 50–80mW [49, 50]. Better results have been reported in academic papers [51, 52, 53], with Pain [53] reporting power consumptions between 10–20mW at 30frame/s for a 512x512 pixel imager.

Computationally, change detection can be reduced to a bare minimum of 6 operations/pixel:

1. Fetch 8-bits serially from ADC.

2. Load this value into the accumulator.

3. Get old pixel value for this location from storage memory.

4. Compare the two pixel values.

5. Store binary result of comparison.

6. Store new pixel value to storage memory.

These 6 operations do not directly translate to processor instructions of course. Overhead to implement loop structures, longer cycles for processor Jump instructions, and general housekeeping in the code will cause the number of instructions/pixel to be significantly higher. 8instr./pixel will be used as a conservative figure to address the extra instructions necessary for overhead. With this figure, even if the number of pixels matches DPCT imager's APS array resolution of 120x36pixels, a total of 34,560instructions/frame would be required just for rudimentary change detection. To operate at 30frame/s, a processor capable of at least 1.036MIPS would be needed.

It is noted that processor efficiencies of 1MIPS/mW and greater have been achieved by various processor architectures. Such processors are usually designed for over 1000MIPS operation and over 1W power consumption. However, what is important for this comparison is not just an efficient processor, but a low-power processor in absolute terms. 1000MIPS is not required for the change-detection task, and we cannot permit power usage to even approach 1W. A survey of processors available at the time of publication shows that the MSP430 series of microcontrollers from Texas Instruments currently has the best efficiency while still allowing operation down to 1MIPS. In fact, the processing efficiency of this processor family is 550μW/MIPS at all possible operating speeds with a 2.2V supply[54].

However, these processors have 256 bytes or less of on-chip SRAM, and very few I/O ports to enable access to off-chip RAM. Even if they could access off-chip SRAM, operation would be much slower. In addition, power consumption would markedly increase because of the additional SRAM chip, and the constant charging/discharging of the I/O pins capacitance during communication with the separate SRAM. Once off-chip SRAM is necessary, the options become numerous. In general, at the present time there is unlikely to be a solution which uses less than 0.6mW/MIPS to carry out the processing required and utilize off-chip memory.

Thus, the traditional change detection method is dominated by the imaging power, using over 10mW for imaging and about 0.55mW for processing. In comparison, running both APS and centroid tracker of the DPCT chip consumes 2.6mW of power. With the centroid tracker alone running, power usage reduces to about 0.6mW of power. Change detection is thus carried out 10 times more power-efficiently than with the traditional method. In addition, more powerful information is available to the consumer of change data—position tracking of moving objects. This enhanced functionality can be used, for instance, to track a moving object rather than just notice activity. Comparable functionality from a microprocessor would use many, many more MIPS and increase power demand even more. Additionally, parallel access to the image plane allows the power consumption of many circuits in the DPCT chip to scale as the square root of total pixels, while a microprocessor-based architecture would necessarily scale linearly with pixels.

For larger imagers the difference in power consumption becomes even more favorable to the DPCT.

4.5 Economical 3-D Human Computer Interaction

As the computing power of desktop home computers increases, 3-D applications have become more numerous. CAD programs, games, data visualization, and virtual reality all require human interaction with a computer representation in three dimensions. The main spatial human input device in use today, the 2-D mouse, is inadequate for such tasks. For closer and more direct interaction of people with these virtual spaces, methods for tracking position and rotation of people and the objects they work with are necessary. Additionally, "Augmented Reality" applications that mix computer-generated images and information with a real-world task require even more precise registration of objects in the real world to the virtual.

Numerous methods exist of tracking objects with six degrees-of-freedom (3 translational axes and 3 rotational axes.) As Welch and Foxlin note, there is no one optimal method for all tasks, but specialized methods that are better suited different specific tasks [55]. A list of the most widely-used technologies for position tracking includes the use of magnetic fields, inertial methods, direct mechanical linkages, and optical methods.

Sensing magnetic fields is one of the most-used methods in the entertainment industry. Human performers are tracked and computer-generated characters are directed to mimic their moves. A magnetic field source, using either AC or quasi-DC fields, generates the field that is either sensed by coils (for AC beacons) or magnetometers (for DC fields). A strength of this method are that it is relatively immune from non-conductive occlusions: magnetic fields can pass through a person, for instance. However, AC magnetic fields are affected by the eddy currents in conductive objects in the sensing area. Also, ferromagnetic objects can affect DC magnetic field measurement. These interference issues are the reason that magnetic position sensing tends to be used in very well-controlled settings, devoid of any objects other than the performer being tracked and the tracking equipment.

Inertial navigation is the practice of sensing angular and translational accelerations to calculate current position. It has been used in ships and airplanes on a macro scale for decades. Large-sized inertial navigation units can use spinning gyroscopes or laser gyroscopes to sense rotation and stabilize a platform on which linear accelerometers are placed. Unfortunately, small-sized inertial position tracking units are too small to use the same kind of actively-stabilized platform. They typically use Micro-Electromechanical Systems (MEMS) gyroscopes and accelerometers to sense angular and translational acceleration. They have no active stabilization, however, so any errors in measurement and calculation accumulate and cause the position estimate to diverge from the actual position more and more with time.

This "drift" is usually quite significant, and is why these sensors are almost never used alone, but instead with other absolute positioning systems.

Methods using a direct mechanical linkage to sense position and rotation are the most direct and probably the easiest to understand. The basic idea is to have the object being tracked connected to a fixed base at all times by a jointed mechanical arm. The angular position of the joints of the arm can be recorded, and together with knowledge of the length and geometry of the arm, the position of the tip of the arm can be calculated. This method is very accurate. It also allows for the possibility of introducing force-feedback to the arm to allow haptic feedback to the user. Unfortunately, the main drawback to this approach is that it is cumbersome. Because a mechanical arm is attached to the object or instrument being used at all times, freedom of movement is limited. The inertia and physical space taken by the arm can be encumbering, preventing natural movement of the object by the user.

The remaining method of position tracking is by use of optical cues. The "holy grail" of optical tracking would be a device that can track position and rotation by looking at natural objects in the environment only. However, most trackers that are commercially available use artificial beacons such as LEDs as landmarks. By selectively turning on or off each LED, an optical sensor can sense its position relative to a particular LED. By using multiple views of the same LEDs, the position and rotation of the sensor with respect to the beacons can be extracted.

The best example of a sensor capable of tracking multiple LEDs with accuracy is the Highball Tracking System [56]. It consists of arrays of infrared LEDs mounted on the ceiling and a single tracker to be mounted on the object being moved. The LEDs are activated one at a time. The tracker unit consists of 6 photosensors with different views of the ceiling. The photosensors have IR filters which remove all visible light, allowing mainly just the light from the infrared LEDs to pass. By using lateral-effect photodiodes, the centroid position of one spot of light can be found. The system works because most ambient light is filtered out, and using the assumption that the IR LED active at any given time is the dominant source of IR in the environment. Using the centroid positions from the photodetectors, and a statistical computational framework, specialized hardware can computer centroids at 1500 positions/second.

The Highball Tracking System is designed for a wide-area deployment, with rows and rows of LEDs mounted on the ceiling to be effective. The specialized hardware necessary to compute positions as quickly and as accurately is very expensive, even with mass production. While the Highball System is optimized for very careful and fast position tracking for uses such as head-mounted displays, the requirements for a desktop 3-D mouse are not nearly as rigorous. For the more limited area of operation of the desktop, a solution based on the centroid tracker may be effective, more economical, and require less hardware on the object being tracked.

A DPCT imager based 3-D mouse would work by using two or more DPCT imagers viewing the same instrument or object being manipulated by the user [57]. The Highball system places the sensor on the tracked object. In a DPCT imager system, only tiny LED beacons need be on the tracked instrument, instead of the 7cm-diameter, 7cm-high sensor cylinder of the Highball System. LEDs are widely available with sub-millimeter dimensions, fitting on very small tools. With very thin wires or even wireless activation, the LEDs can blink in a sequence that will allow two cameras to locate the LED in 3-space using simple stereo vision techniques. The math to find depth from displacement is undemanding, and any more cameras would serve to increase the accuracy of the depth measurement, as well as guarding against occlusion of view of the beacons by the cameras. With 3 LEDs, a full 6DOF position of the instrument or object could be found.

The opportunity exists to use the tracker in a more active position-sensing method. By mounting the tracker on a Pan-Tilt base for the camera, an active tracker could attempt to keep one beacon in the center of the field of view. This way, a larger area could be viewed by more precision, because the optics for the camera could be made more telephoto, enlarging the view of the beacon. The angles of the Pan and Tilt (altitude and azimuth) would be the coarse measurement of the direction of the beacon, and the position noted by the tracking imager within its field of view would add the fine angle information. By keeping the beacon near the center of the field of view, nonlinear effects such as lens distortion could be

kept to a minimum, while the relatively linear and accurate rotational position of the camera mount would allow for good position measurement over a wide area. With this method, a wide-area tracker with good precision may be possible.

Because the DPCT imager also includes a full APS imager, real-time video can also be obtained with little extra cost. For person tracking, the added value of live video is obvious. In addition, more complicated processing such as gesture recognition or object identification could be carried out in the PC. With the combination of ultrasound and sensor fusion techniques, even more 3-D spatial information about the scene is available.

4.6 Next Generation Imager

The general concept of the DPCT imager is to use an application-specific imager and processing to increase efficiency. By optimizing the design for specific tasks, \irrelevant data can be eliminated and power consumption reduced. As stated before, the DPCT imager uses temporal changes or the absence of change in the scene to make decisions about the relative importance of the data. For a stationary camera view with moving objects, most of the pixels will not change in time and will be considered "unimportant". By concentrating on the other dynamic pixels, processing resources can be allocated where they are most needed.

Reducing this idea to its most basic kernel, the DPCT is an imager that performs

data compression based on movement. If one is only interested in reduction of data, the essence of this method provides an excellent reduction of data with very little computation. In fact, one could call this type of compression "Data Triage", because only pixels that are important are allowed to use limited communication and processing resources. Mathematical transformations are not being carried out on the entire image, but rather decisions are being made that gate which parts of the image receive attention. It is a basic metric, but one that lends itself well to extremely efficient coding of the image, and integration into the image sensor itself.

Compression on the imager was first implemented in the most way: by placing formerly separate imager and digital compression units on the same chip. The original appeal of APS technology was that formerly separate imaging, digitization, and processing units could be put on the same piece of silicon to make a "Camera-on-a-Chip" [42]. As engineers became more comfortable mixing pixels and processing, they were naturally led to closer coupling of image reception and image processing blocks. One approach that emerged is to process multiple pixels in parallel as they are read out (row-by-row, for example) [58, 59, 60]. In the strictest definition of "focal-plane processing", processing circuitry is actually placed inside of the imaging pixel array. This approach can be thought of as the ultimate intertwining of image sensors and processing, and has been traditionally used in computer vision applications where feature extraction takes priority over

image quality [61]. However, there were some significant attempts to construct an imager with focal-plane processing for high-quality imaging [62, 63, 60]. These notable examples offer methods for image compression. They use parallel input data directly with parallelized compression algorithms, eliminating the inefficient serialization of data required by most imaging systems. Leon [60] implemented a spatial compression, producing data only when a pixel differs significantly from its spatial neighborhood. Aizawa [62] implemented a chip with per-pixel processing to extract temporal changes and conditionally output only those pixels that changed since the last frame. Unfortunately, Aizawa's implementation uses a very large pixel. From the lack of subsequent articles on the architecture, it appears as if this may have been a very practical problem which hindered further development.

Leveraging the low-transistor-count design of the DPCT centroid-tracking pixel, a next-generation chip has been constructed by Mallik, myself and others, using a similar topology but oriented toward compression based on temporal change [1]. This type of compression is called conditional replenishment [64]. The new chip used a more hybrid approach than Aizawa, adding a small amount of the circuitry to the pixel, but using circuitry outside the array as much as possible to do the bulk of computation on readout. The benefits of this are: circuit elements do not have to be duplicated for each pixel, extra power for these computation circuits is only drawn when the pixels are accessed, and the fill factor of the array is also increases. Of course, computation on readout decreases the amount of continuous parallel

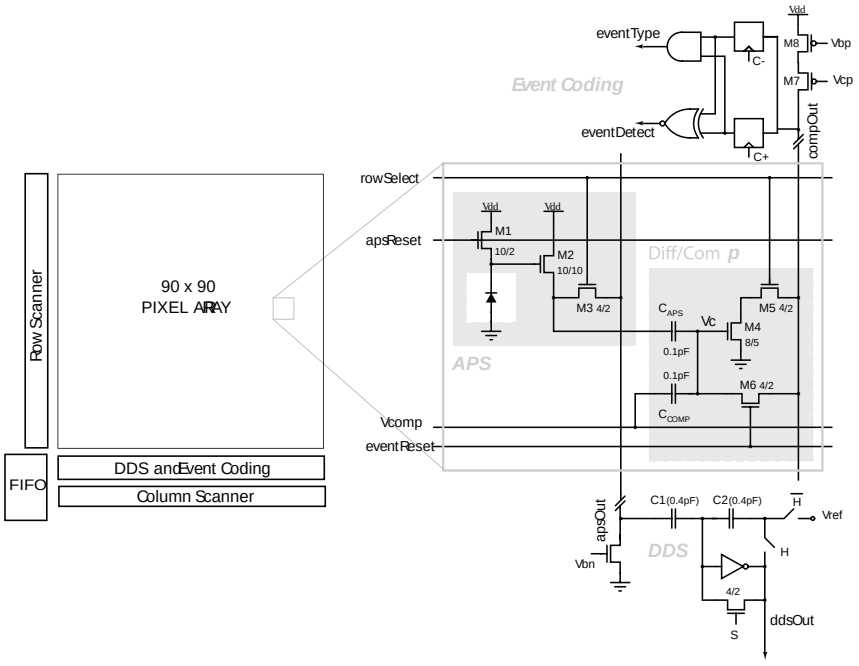


Figure 4.1: Schematic of the Temporal Threshold Change Detection Imager. Detail is pixel with selected edge circuitry. (Courtesy U. Mallik [1])

processing possible, but for this application such processing isn't necessary. Because this compression is based on the temporal response of each individual pixel and not its neighbors, there is no need for all pixels to compute simultaneously.

The pixel schematic is shown in Figure 4.1. The pixel uses 2 explicit capacitors and 6 NMOS transistors. This is far less than the 29 pixels, both NMOS and PMOS, and two explicit capacitors used in the design by Aizawa. Each pixel in the Temporal Threshold Change Detection Imager (TTCDI) is 25.2μm $\times$ 25.2μm, whereas the previous pixel by Aizawa was 170μm $\times$ 170μm. Of course the process used for the Aizawa chip had a larger feature size. To compare both pixels designs using

their respective scalable rules, $\lambda = 1\mu\mathrm{m}$ is used for Aizawa and $\lambda = 0.35\mu\mathrm{m}$ for the TTCDI. With lambda-based rules and given the same process, the pixel of the TTCDI still would use a mere 18% of the area of the older pixel.

The temporal detection of the TTCDI occurs during readout. At the end of an APS voltage read, the APS integration voltage is stored on capacitor C_{APS} in a similar manner as the current light level was in the centroid-tracking pixels of the original DPCT imager. The inverter is reset, setting input voltage equal to output voltage at the trip point of the inverter. The difference between the trip point and current APS voltage being read out is then stored across C_{APS}. When the same pixel is being read on the next frame, if the new APS integration voltage is higher or lower than before, the input of the inverter will be higher or lower than the trip point, causing the output of the inverter to go low or high, respectively. The ingenious addition of capacitor C_{comp} adds additional flexibility to this scheme. By changing the voltage V_{comp} to be slightly higher or lower than the state it was in when the last APS integration voltage was stored across C_{APS}, the threshold can be moved to be a fixed amount higher or lower than the old threshold. This allows testing the output of the current APS voltage against multiple voltages related to the original APS voltage. This capability is used to implement true change detection with a threshold for change. By comparing the $V_{\mathrm{APS}}(\mathrm{frame})$ against $V_{\mathrm{APS}}(\mathrm{frame}-1)+\Delta V_1$, and $V_{\mathrm{APS}}(\mathrm{frame}-1)-\Delta V_2$, three regions are defined: significantly lower voltage, not significant voltage change, and significantly higher

voltage. What amount of change is "significant" can be controlled by the value of ΔV_1 and ΔV_2. These external control voltages allow for a flexible definition of light intensity change. The information can be used from these two comparisons to gate the transmission of the current pixel value. If no change is detected, the chip doesn't bother to transmit and convert that pixel to digital. If it does see significant change, the pixel is reported.

Not only is power conserved that would be spent in an ADC, but the power used by the drivers to transmit precise analog or fast digital data off-chip is eliminated also. Thus, within the same time and power budget, a much larger array can be read because it is unlikely that 100% of the pixels will need to be transmitted from each frame. In fact, for a stationary camera and head-and-shoulders view of a person, typically less than 10% of the pixels will need to be updated in a given frame [65].

In addition to change-detection, the Change Threshold chip is flexible enough to allow for other sophisticated modes of operation. If instead of sampling the previous APS voltage the original reset voltage is sampled instead, true Correlated Double Sampling is possible. Only when the reset voltage is sampled before the final integration voltage is the thermal noise of the two voltages truly correlated. Normally, as an expedient shortcut the reset voltage is sampled after the final integration voltage. This imperfect double sampling corrects for 1/f noise and fixed-pattern noise only. Because the TTCDI can store an extra voltage for every pixel,

this storage can be used to keep a copy of the original reset voltage and execute true correlated double sampling.

Also, because of the flexible method of adjusting the threshold of the comparison, an arbitrary number of comparisons can be executed using multiple V_{comp} voltages. Using the original reset voltage as the first stored voltage, we can execute any number of separate comparisons, resulting in binary output data for a pixel-level Analog-to-Digital conversion. This operation has been demonstrated to work at 6 bits per pixel in laboratory tests.

4.7 Summary

From these examples, we can see the major advantages of the DPCT Imager demonstrated. Primarily, it provides a power efficient way to carry out imaging and computation. It also allows speed improvements, including increased throughput and near-zero latency. Because of this dual quickness, the imager can be used in creative applications where visual sensing is placed within a feedback loop. The low-latency, near-instantaneous operation is an especially important advantage over traditional computation methods, preventing instability and allowing faster loop speeds.

The keys to these advantages are direct access to the imaging array and application-specific processing optimized for change-detection. By bypassing the serialization

and analog-to-digital conversion parts of traditional imaging process, efficiency of computation is boosted. A next-generation chip using the same techniques for video compression has been constructed. Using similar design methodology, it has also demonstrated high-performance operation with frugal use of resources.

Chapter 5

Sonar Bearing Angle Estimation Using Spatiotemporal Frequency Filters: Theory and System Architecture

5.1 Introduction

Shortcomings of traditional array processing techniques have been detailed in Chapter 1. To address these shortcomings, I have developed an approach for array processing which is optimized for the task of determining the location of the signal source. This method, spatiotemporal frequency filtering, utilizes a bank of

bandpass filters in space and time. Each spatiotemporal filter includes one spatial bandpass filter followed by one temporal bandpass filter. When applied to array signals, each filter is sensitive to a particular wavefront velocity determined by its temporal and spatial bandpass center frequencies. As will be shown, the velocity of a wavefront across the array is directly related to the angle of incidence of the incoming signal to the area. Different filters can be designed to be sensitive to different wavefront velocities and hence different incident angles. An ensemble of different spatiotemporal filters can be examined in parallel to determine which filter has the maximum response, and consequently the incoming bearing angle of the input signal. By only utilizing simple bandpass filters for the bulk of the processing, the digitization step of most traditional array processing can be skipped entirely, reducing power and system complexity. I will show that I can use this configuration to realize my design goal of a system capable of 1° accuracy. The chip I designed that accomplishes all of this processing is the Spatio-Temporal Frequency Array Processor, or STFAP.

In addition, because the system can use any zero-crossing of the carrier wave for wavefront velocity extraction, it is not bound to making just one observation per sonar pulse. This differs from single-sensor or array Time Of Flight (TOF) sonar systems, which only observe the time of the onset of the pulse envelope of an ultrasonic return. In my system, multiple observations are possible from the same pulse at different times. In a single object case, this can improve accuracy by

using redundant measurements during the length of a sonar return. In a multiple object case, different returns from the same output sonar ping can be analyzed separately to extract bearing directions to all objects. As long as the objects have different ranges from the sensor, their echoes will be separated in time and thus able to be processed in sequence as they return to the sensor array. It takes some time for an object's wavefront to move from one end of the array to the other, and two pulses cannot be measured on the array at the same time. Consequently, the necessary object depth separation is directly proportional to the length of the array. A smaller array, such as one based on MEMS technology, would actually help make this requirement less stringent, allowing objects to be more closely-spaced in depth.

Most regular heuristically- or statistically-processed arrays, (including those referenced in this thesis,) process information based on the range information of their ultrasonic sensors only, and would thus be confused by multiple objects. If they were able to process them as individual objects, they would still need separate pings to analyze each separate object. Because my system is able to extract more information from a single ping, the number of pings to image an environment is lower than other methods. This has an especially beneficiary impact on power consumption, due to the relatively high power needed to broadcast and successfully receive an ultrasonic burst in an attenuating medium such as air. Fewer ultrasonic bursts always take less power than more bursts.

5.2 Spatiotemporal Theory

Spatiotemporal frequency filters are also known as *velocity filters*, because they can be tuned give response for particular *velocities* [61]. Velocity filtering as a concept comes from study of the retinal vision processing in certain mammals [66], where such processing is believed to take place. In this case, the velocity refers to the projection of object images passing over the retina focal plane. The earliest spatiotemporal-frequency filtering models were constructed not only for functionality, but also to be biologically-plausible processing systems [67, 68, 69].

The microphone array of my system is not looking for moving objects—it is looking for moving wavefronts. A signal wavefront will appear to be moving across the array if the source of the signal is from a direction closer to one end of the array than the other. Even though the signal source is stationary, the emitted signal will appear to cascade across the array with some velocity. The value of this velocity can be extracted to find the bearing angle of the original signal source to the array. Simple geometry and Fourier theory are all that is needed to understand the processing involved.

The time delay between the signal reaching one sensor and the previous sensor is equal to the extra path length divided by the speed of sound, or

$$t_{\text{extra}} = \frac{x_{\text{extra}}}{v_s} = \frac{x_0 \sin \theta_{\text{incident}}}{v_s} \tag{5.1}$$

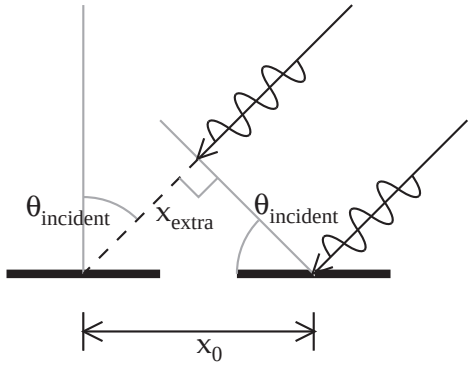


Figure 5.1: Two sensors receiving signal copies of the same far-field source.

Here v_s is the speed of sound, x_{extra} is the extra path length traveled by the wave, θ_{incident} is the incident angle off normal to the array, (which can be positive or negative,) and x_0 is the inter-sensor spacing. See Figure 5.1. The apparent velocity across the array due to a certain incident angle can then be derived:

$$v_{\text{array}} = \frac{x_0}{t_{\text{extra}}} = \frac{v_s}{\sin \theta_{\text{incident}}} \tag{5.2}$$

For shallow angles, (90° off-normal,) the apparent velocity is minimum, and equal to the speed of sound. At an angle normal to the line of the array, (0°,) velocity appears infinite.

The spatiotemporal method for determining velocity is easiest to understand when starting from a space-time representation of a moving object. Figure 5.2 shows that a moving object will trace a line in a graph of space and time. The velocity of the object (x/t) will be equal to the slope of the line in this representa-

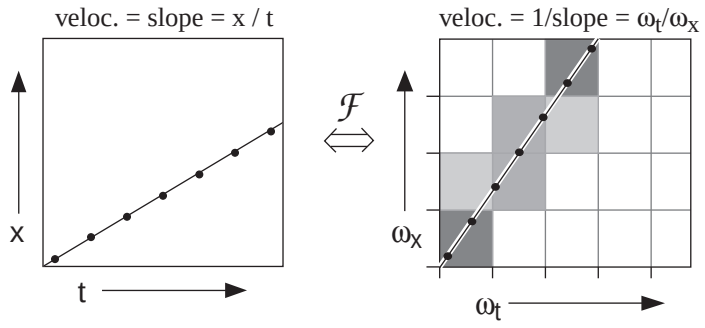


Figure 5.2: Velocity represented in space-time and frequency space-time. Illustrated is the response of spatiotemporal frequency filters sensitive to bands (regions) in the ω_x-ω_t plane.

tion. If we suppose this object is an impulse, (or delta-function,) and take a 1-D Fourier transform in space or a 1-D Fourier transform in time, the temporal frequency spectrum or the spatial frequency spectrum will be perfectly flat. Equal energy will be present at all frequencies. However, more interesting behavior can be noted when a full 2-D spatial/temporal Fourier transform is performed. Energy is not distributed evenly over the whole ω_x-ω_t plane. In fact, an impulse tracing a straight line in the x-t plane will also have energy over only a straight line in the ω_x-ω_t plane. [70] shows a derivation of a similar relationship: the three-dimensional (2-D spatial and time) Fourier transform of a plane in order to illustrate spatiotemporal frequency spectrum of movement in 2-D images.

The inverse of the slope of the frequency-domain line, or ω_t/ω_x, is also equal to the velocity of the impulse. By cascading spatial- and temporal-frequency bandpass filtering, spatiotemporal filters can be constructed sensitive to specific areas

of the ω_x-ω_t plane. It is possible to use spatiotemporal filters with separable spatial and temporal responses, reducing each filter to a spatial bandpass filter cascaded with temporal bandpass filter. Using these filters, it is possible to sense if a velocity line is passing through a particular region.

This approach is advantageous for some important reasons. The effects of noise are reduced because each filter responds to a limited frequency range, which limits the total energy of received noise. Highpass filtering, for instance, has a well-known sensitivity to noise because it responds to all high frequencies, even those out of the useful signal band. In addition, multiple spatiotemporal filters whose areas of sensitivity overlap the line of movement will each give a response. By considering the ensemble of filter responses, an estimate of the velocity can be extracted as a weighted average of the filters involved. This helps the combined system to be more robust. It has been shown [71] in simulation that this approach can tolerate moderately low signal-to-noise ratios, which is demonstrated in Figure 5.3.

The same technique can be used with other objects besides an ideal impulse, but the resulting spatial and temporal spectra may not be flat, and may even be band-limited. In these situations, extra care is necessary when decoding the output of the filters. Amplitudes of the filters may be different than expected, or non-existent if the spectrum of the object has no energy in the spatial or temporal frequency-band being measured.

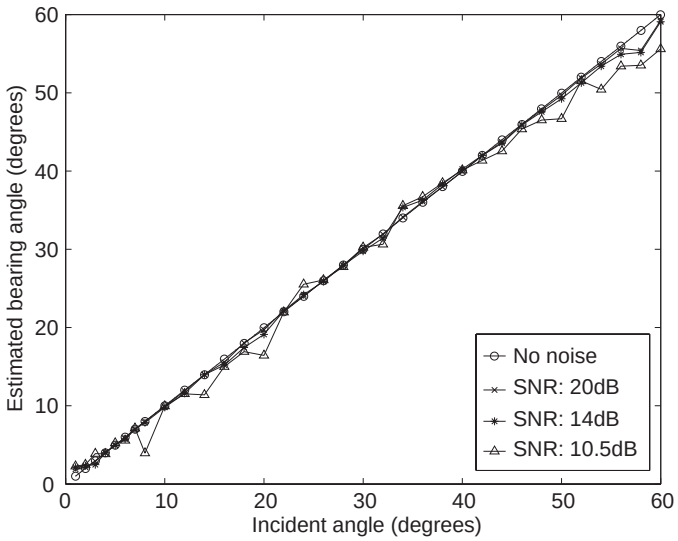


Figure 5.3: Simulation results showing actual angle/reported angle relationship in the presence of different quantities of random noise.

In my target application, I are looking for the angle bearing of a short burst of an ultrasonic frequency. It is convenient to convert the modulated carrier into a plain square pulse by special preprocessing described in Section 5.3.3. A square pulse does not have a flat frequency spectrum, but its spectrum is broad enough that some energy will be present at all temporal and spatial frequencies within range of our processing.

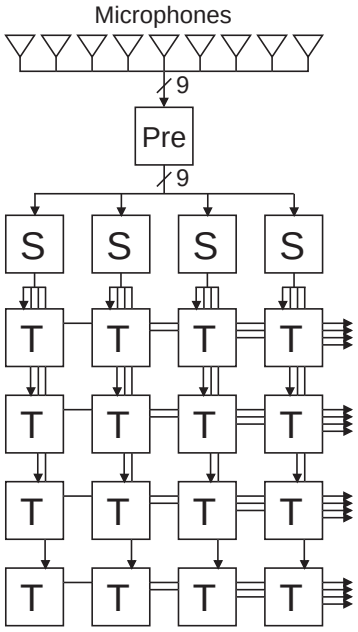


Figure 5.4: Block diagram of the entire system. The block labeled "Pre" is the preprocessing block, "S" denotes spatial filters, and blocks labeled with "T" are temporal filters.

5.3 System Architecture

5.3.1 System Overview

A block diagram of the entire STFAP system is shown in Figure 5.4. Starting with the microphone array, all signals pass through the preprocessing block and are copied to each spatial filter. The output of each spatial filter is then copied to four temporal processing blocks, which leads to 16 separate outputs from the entire system.

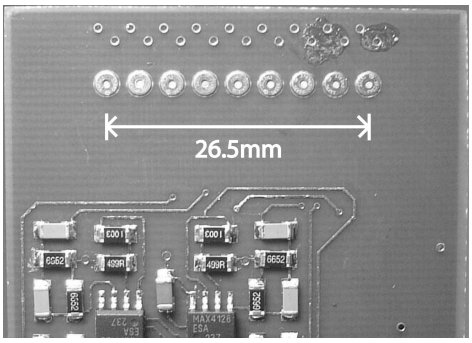


Figure 5.5: Photo of microphone array module.

5.3.2 Microphone Array

At the beginning of an array processing system are the actual sensors. My microphone array module consists of a printed circuit board with mounted 2.5mm-diameter microphones, and simple two-pole highpass filters with amplification following each microphone to remove audio frequencies and boost the signal. The microphones themselves have an intrinsic acoustic frequency limit, causing response to roll-off at high frequencies. There are nine microphones, with 3.3mm inter-microphone spacing and a total combined baseline of 26.5mm. This is a small array, but I envision applying this processing to much smaller arrays. A goal of the project is to miniaturize the microphone array to the point where the entire array can be one MEMS device, which will improve uniformity of microphone structure and response, as well as taking up much less space.

5.3.3 Preprocessing

As mentioned in the introduction, the time resolution demanded to accurately measure bearing on a small-baseline requires sub-microsecond precision. Varying gain and phase-delay of the environment, microphones, demodulation, and amplifier circuits make pulse envelope detection a highly error-prone way to detect sub-microsecond delays. In addition to these errors, the carrier frequency itself poses a problem: to reduce attenuation, a near-audio frequency of 40kHz was chosen. The small available bandwidth at this low frequency does not allow a wide-band square pulse to be modulated on it as a pulse envelope. Only more slowly-varying, band-limited shapes are available, making envelope-detection trickier.

Instead of sensing the delays in the edges of the sonar pulse, a more synchronous approach was found to meet the demanding time requirements. The current method is similar to existing phased array techniques by using the zero-crossings of the array signals as time references. Digital square waves are generated from the zero-crossings of the received ultrasonic input signals. These digital square waves could, in theory, be sent directly to the spatiotemporal processing unit. They would appear as a train of pulses, which by superposition would be as valid to a linear filtering system as a single pulse is. However, the implementation details of my spatiotemporal filtering system as described in Sections 5.3.4 and 5.3.5 necessitate that only one pulse at a time be "visible" on the array at a time. If one array input is displaying the digital "high" of a pulse, the other array

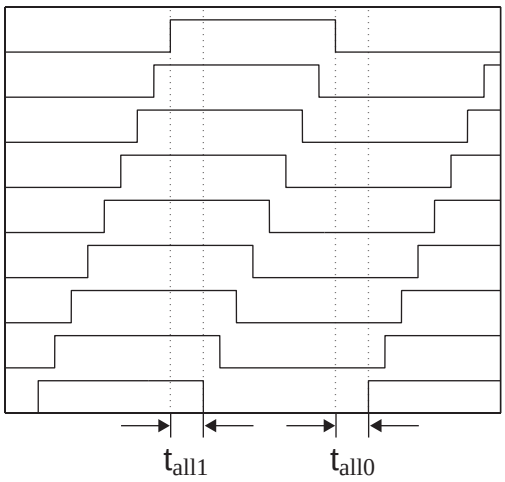


Figure 5.6: Ensemble of pulses are required to have $t_{\text{all1}} > 0$ and $t_{\text{all0}} > 0$.

inputs must either be relaying a digital "high" of the *same* pulse or show no activity. Figure 5.6 illustrates the equivalent requirement that there be some amount of time when all array inputs are "high," and some amount of time when all array inputs are low. Because of this requirement, the last part of the preprocessing system generates widely-spaced long pulses from the accurate timing information of the square wave zero-crossings. The intermediate square waves themselves have too short of a period to meet these timing restrictions. Figure 5.7 shows the transformation of a raw microphone input waveform to a long pulse output.

The zero-crossing detector used in the testing of the STFAP is a simple threshold circuit, comparing each microphone's input signal with a fixed zero voltage. This method produces a square wave output that has the same zero crossings as the original sine wave input. It is a basic circuit implemented off-chip, to enable char-

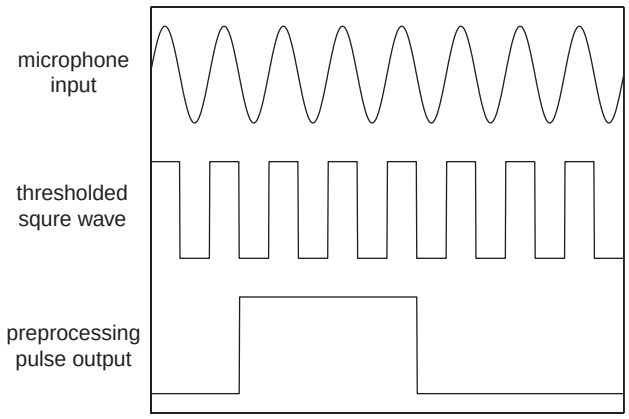


Figure 5.7: Preprocessing waveforms.

acterization of the current chip. Future chips will have a more sophisticated zero-crossing detector on-chip. The planned circuit includes adaptive "zero" threshold and a separate signal-strength threshold, to disable zero-crossing detection for very small signals or noise.

Fulfilling the requirement that there be only one pulse visible to the array at a time is not as trivial as it might appear on first glance. The procedure can be summarized as follows:

1. Identify the edges in all sensor responses that correspond to the same edge of the signal source. Collectively, I'll call these edges "correlated edges".

2. Start the output long pulses from these edges.

3. Wait.

4. Finish the output long pulses on correlated edges.

5. Wait.

To begin with, the relevant inter-element delays for the array must be discussed. The longest delay between two microphone signals occurs when the incoming sound is a full 90° from the array normal. At this bearing angle, the incoming sound is travelling in line with the line of the array, and the inter-sensor signal delay is just the inter-sensor distance divided by the speed of sound. For adjacent sensors 3.3mm apart, this delay comes out to 9.7μs, and the delay for sound to travel the entire length of the STFAP array is 77.9μs. The period of the utilized 40kHz ultrasonic frequency is 25μs. It's important to note the two maximum delay times with respect to this ultrasonic period. First, correlated edges of adjacent sensors will be delayed less than half of a full input period. This means correlated edges on two adjacent sensors can be identified, because it is known that a rising (falling) edge on one sensor corresponds to the closest in time rising (falling) edge on an adjacent sensor. In contrast, the response of sensors at the ends of the array can be delayed over three full periods of the input signal. So viewing two sensors at opposite ends of the 26.5mm array and identifying correlated edges is impossible with a 40kHz waveform. The delay could be anywhere between 0–3 wavelengths long, depending on the bearing angle.

To make the long pulses output by the preprocessing match the input waveforms in sequence and timing, I have created a simple digital system. The system enforces local sequencing by matching correlated edges on adjacent sensors. When

all local conditions of the overlapping pairs of adjacent sensors are combined, the nine output pulses can be correctly queued so that they all begin on correlated input edges. This ordering logic eliminates the problem of the unknown number of wavelengths between the two farthest sensors, because it enforces local ordering for all adjacent pairs in between these two sensors. The output pulses, once properly begun, merely count a number of input wavelengths for the logic "high" duration, and a number of input wavelengths for the following logic "low" duration. In this way long pulses are generated that have the exact same timing as the input signals. All fine timing such as noise-induced jitter and other artifacts are preserved. The only thing this part of the preprocessing does is ensure that output waveforms are sequenced correctly and occur in same order as correlated edges appear to its inputs.

The ordering subsystem was developed and is implemented on an FPGA. The FPGA provides a rapid means to prototype algorithms and test them in a real-world environment. Now that a robust and effective processing strategy has been determined, integration of this circuit on-chip is the next step in development. The logic is amenable to non-clocked asynchronous operation, which will allow for lower-power operation and compact implementation.

5.3.4 Spatial Filters

The spatial filters accept continuous-time, binary data from the nine sensors, perform spatial bandpass filtering, and provide continuous-time, discrete value outputs. Inputs are in the form of digital voltages from the preprocessing unit, and the output is a variable current.

A linear array of microphones inherently represents space discretely: the length of the array is being sampled with only nine sensors. As such, the filtering operation immediately lends itself to discrete data processing techniques. To perform a spatial bandpass, the nine inputs are convolved with a kernel. The chosen a kernel has nine coefficients, the same length as the number of inputs. Because the number of coefficients matches the length of the inputs, only one value is output, and it is also not necessary to apply the convolution kernel to different positions of the input. Thus the entire convolution reduces to multiplying each input by its respective kernel coefficient, and summing the results. It is simplified to a vector multiplication.

The discrete-space filters, shown in Figure 5.8, are constructed of nine variable current sources, with nine switches. The current sources are bidirectional current-DACs. Each coefficient value is loaded onto a separate current-DAC, and the corresponding digital input from the array switches the current produced by that DAC on or off a shared node. The current-DACs and switches perform multiplication of input binary data to integer coefficients, and the shared node sums all currents.

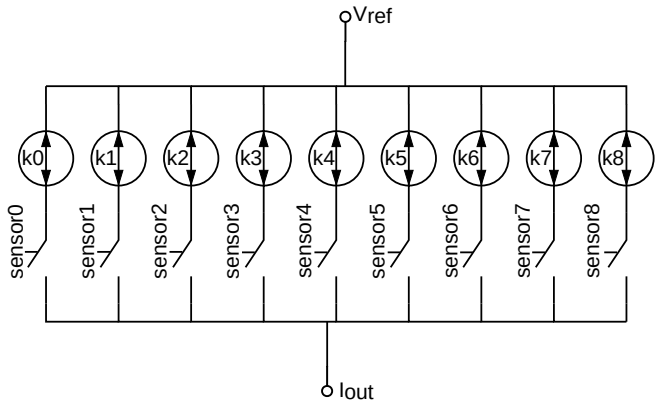


Figure 5.8: Spatial filter schematic.

The output is represented as a continuous-time, discrete-level current.

Each current-DAC is represented as a series of binary-weighted currents mirrored off the same reference current. Each DAC bit switches its respective current onto the DAC output if enabled.

5.3.5 Temporal Filters

Spatial bandpass filtering isolates a spatial frequency range. To select regions of the 2-D ω_x-ω_t plane, it is also necessary to filter for a particular temporal frequency range. This can be accomplished with temporal bandpass filtering. With enough spatial and temporal bandpass combinations, the entire ω_x-ω_t plane can be spanned by individual regions.

Instead of isolating specific 2-D regions, I first isolate a strip of the ω_x-ω_t plane corresponding to a spatial band, as described above. Then, instead of simply fil-

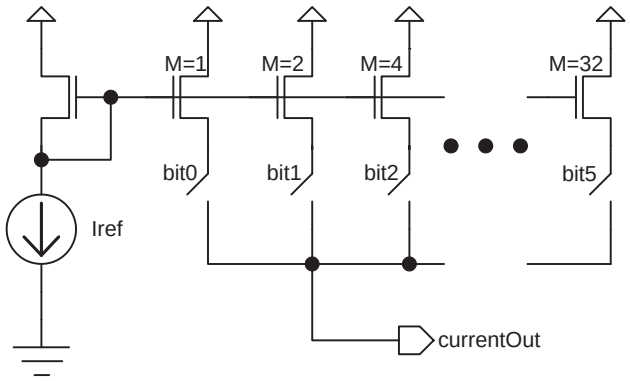


Figure 5.9: Current-DAC schematic. (Sign circuitry not shown.)

tering again for temporal frequency discrimination, I have implemented a circuit which performs a frequency-to-voltage function in the time domain. The circuit that performs this is a simple integrator which is first reset, starts integration at the first activity on the array elements, and ends after all activity on the array has ceased. In this way, an estimate of the location of the velocity line is generated in frequency space from each spatial/integrator pair, instead of merely an ensemble of filter responses which would require additional processing layers to extract velocity. In addition, an integrator is a much more compact and simple circuit than a full temporal bandpass circuit. Thus silicon area is conserved and complexity in the system is reduced, aiding system robustness.

An integrator performs frequency-to-voltage conversion in this case because of the conditions imposed by my preprocessing and spatial filters. An additional constraint is also necessary: to restrict the spatial kernels to those that have no

response to 0 spatial frequency (all sensors high or all sensors low.) Because the output signal lacks any DC offset, the spatial current output current must also be rectified so it doesn't integrate to 0. Within these constraints, a change in the velocity of an edge across the array will only cause time dilation or contraction of the spatial filter output during the passing of the edge. During the middle of the pulse, all array inputs are high, and the spatial filter responds with 0 current. f is defined as a generalized function describing the basic shape of the waveform produced by a passing edge on the array, due to the spatial filter and preprocessing. The duration of the passing pulse will be inversely proportional to its velocity passing over the array. $t = 0$ is defined as the beginning time of a pulse starting to move across the array. Since integration only occurs during the time an edge is passing over the array, the end time of integration $T_{\text{end}} \propto 1/v_{\text{array}}$. In the ideal case, $\omega_t = \omega_x v_{\text{array}}$, where ω_x is the spatial center frequency after spatial filtering, so additionally $T_{\text{end}} \propto 1/\omega_t$. This mathematical structure allows us to use simple integration to extract the velocity.

If $g(t)$ is the integral of $f(t)$:

$$\int f(t)dt = g(t) \tag{5.3}$$

then with arbitrary constant B denoting the multiple of $1/\omega_t$ that is T_{end},

$$\int_0^{B/\omega_t} f(\omega_t t)dt = \frac{1}{\omega_t}\left(g(B) - g(0)\right) \tag{5.4}$$

Substituting the constant D for the expressions containing $g()$:

$$V_{\text{int1}} = \int_0^{T_{\text{end}}} f(\omega_t t)dt = \frac{1}{\omega_t}D \tag{5.5}$$

From Equation 5.5, I qualitatively argue the inverse relationship between the temporal frequency and the final output of the integrator. For any ω_t, the output of the integrator is $(1/\omega_t)D$, where the constant D remains the same for any ω_t. A more thorough derivation of the precise relationship will be shown in Subsection 5.4.3. The output of the integrator thus gives a value corresponding to the initial velocity, because bandpass filtering for a particular spatial frequency band has already occurred, and fully separable spatial and temporal filtering is used. Example waveforms of spatial output to final integration are shown in Figure 5.10, as would be the response to a single pulse cascading across the array such as one in Fig. 5.6.

Temporal integration on the STFAP chip was performed for each spatial filter with four separate capacitance values. A copy of each spatial filter's current output was supplied to each different integrating capacitance. Capacitance values of 1.5pF, 3.0pF, 6.0pF, and 12.0pF were used. The smaller values enabled more precision, and the larger values could integrate longer signals without saturating. The range of integrating capacitances allowed a wide range of input signal speeds, so that the full range of angles could be processed.

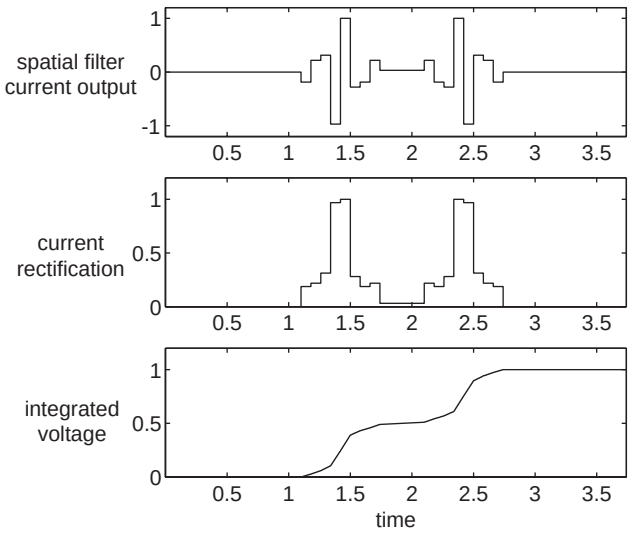


Figure 5.10: Simulated waveforms: spatial filter output (from bandmedhi kernel), rectified copy, and final integrated voltage signal.

5.3.6 Sign detection

The spatial filters are even kernels, (they are symmetric,) the output of these spatial filters is rectified before integrating. Due to these actions, all sign information is lost from the analog signal and a separate circuit is necessary to detect the sign of the input bearing angle with respect to the array. The sign detection circuit is implemented as a simple spatial filter with an odd kernel function. The kernel (-1,-1,-1,-1,0,1,1,1,1) is used. It is followed by a current sign-detection circuit that outputs a digital high or low depending on the direction of the current from the spatial filter. This current sign-detect circuit's value is latched after all sensors show no activity. A pulse moving across the array that ends on one end of the array will cause a negative current to be last output from the odd spatial filter. Likewise,

if the pulse is last seen on the other side of the array the spatial filter will output a positive current last. The sign detect circuit will latch its output of 0 or 1 for each respective case.

5.4 Theory of Operation

5.4.1 Velocity Determination

The overall goal of the processing system is to find bearing angle by extracting the velocity of a wavefront across the array, or in other words, the array sonic flow. The relationship of wavefront to velocity to angle is not linear, and in fact approaches infinity toward 0°, and the speed of sound toward 90° off-axis. Near 0°, velocities rapidly increase, reducing necessary precision in this region, while the slope of velocity change near 90° is much shallower, causing more precision necessary to attain my goal of 1° of accuracy. The precision necessary for 1° accuracy at each bearing angle is shown in the graph of Figure 5.11, expressed in percentage accuracy (velocity at that angle divided by velocity change over 1° at that point.) As can be seen, the same velocity precision as off-axis angle increases yields diminishing returns. As such, I limit the metrics of success to an angular range between -60° and 60°.

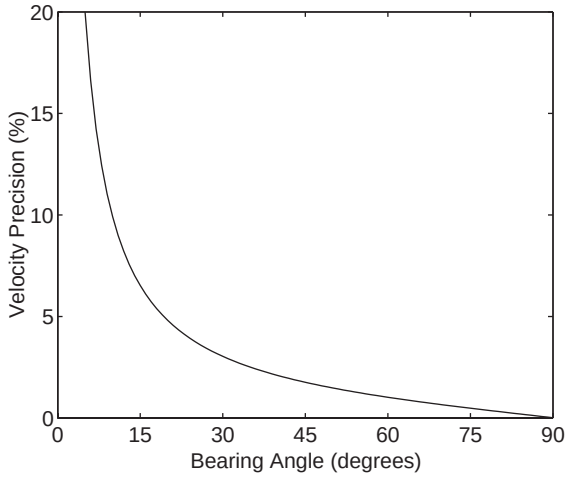


Figure 5.11: Velocity precision necessary for 1° precision at each bearing angle.

5.4.2 Spatial Filtering

I start by choosing appropriate spatial filter kernels. Three of my kernels are kaiser-windowed bandpass filters, and one is a highpass filter. The center frequencies are roughly equally-spaced. The coefficients can be seen in Table 5.1, and the resulting Discrete Fourier Transform plots are visible in Figure 5.12. It should be noted that at least in the ideal cases, all of these filters have virtually no response at 0 spatial frequency. In addition, to use the full resolution of my filters, the coefficients are normalized so that the maximum coefficient for each filter is 63. Consequently, all spatial outputs are not normalized: the "bandhi" filter, because it is a highpass and not a bandpass filter like the others, ends up having twice the output response. Such differences are easy to scale in later stages if direct comparison among the spatial filters is desired.

Filter Name	Coefficients
bandlo	(2 -17 -32 17 63 17 -32 -17 2)
bandmedlo	(3 14 -35 -14 63 -14 -35 14 3)
bandmedhi	(-6 13 3 -41 63 -41 3 13 -6)
bandhi	(5 -19 39 -56 63 -56 39 -19 5)

Table 5.1: Spatial filter kernels

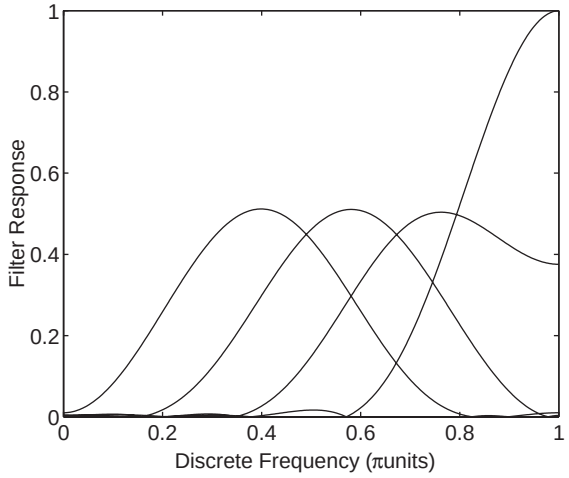


Figure 5.12: Discrete Fourier Transform plots of spatial filter transfer functions relative to each other. Peaks are at 0.4, 0.58, 0.76, and 1 in π-units. 1 π-unit corresponds to the spatial Nyquist frequency.

Because the array spacing is v_0 =3.3mm, the rightmost tick-mark of Figure 5.12 corresponds to a spatial frequency of $\frac{1}{2}(\frac{1}{3.3}\mathrm{mm}^{-1})$ or $152\mathrm{m}^{-1}$. This sets the center frequencies of the four bandpass filters at 61m^{-1}, 88m^{-1}, 115m^{-1}, and the highpass filter is at the top of the range at 152m^{-1}.

5.4.3 Temporal Integration

V_{int} is defined as the final voltage from integration of spatial filter output of a passing pulse. In the ideal case, the subsequent delay between each edge on any two adjacent sensors should be identical. As an edge passes over the array, sensors reading logic-high will increase in number starting from one end of the array and progressing to the other side. An example: sensor 0 is logic-high, then sensors 0 and 1 are logic-high, then sensors 0 and 1 and 2 are logic-high, etc. The process continues until all sensors show inputs logic-high at time t_{mid1}. Then at time t_{mid2} the trailing edge of the pulse will pass over the array, and sensor inputs will go logic-low one by one starting from the same edge of the array. For example: first sensor 0 would go low, the sensors 0 and 1 would be low, then sensors 0 and 1 and 2, etc. This continues until all sensor inputs are again logic-low. The current output by the spatial filters during a passing pulse is:

$$i_{\text{spatial}} = \begin{cases} i_{\text{onset}}(t) & \text{if } t < t_{\text{mid1}} \\ I_{\text{mid}} & \text{if } t_{\text{mid1}} < t < t_{\text{mid2}} \\ i_{\text{offset}}(t) & \text{if } t > t_{\text{mid2}} \end{cases} \tag{5.6}$$

$$\begin{aligned} i_{\text{onset}}(t) &= a_0 I_{\text{ref}} u(t - T_{\text{on0}}) + a_1 I_{\text{ref}} u(t - T_{\text{on1}}) \\ &\quad + a_2 I_{\text{ref}} u(t - T_{\text{on2}}) + \ldots \\ i_{\text{offset}}(t) &= a_0 I_{\text{ref}} u(-t - T_{\text{off0}}) + a_1 I_{\text{ref}} u(-t - T_{\text{off1}}) \\ &\quad + a_2 I_{\text{ref}} u(-t - T_{\text{off2}}) + \ldots \end{aligned}$$

and the corresponding voltage integrated on capacitor C_{int}:

$$V_{\text{int}} = \frac{1}{C_{\text{int}}} \int |i_{\text{spatial}}| dt \tag{5.7}$$

$$V_{\text{int}} = \frac{1}{C_{\text{int}}} \int |i_{\text{onset}}| + |i_{\text{mid}}| + |i_{\text{offset}}| dt \tag{5.8}$$

Here i_{onset} describes the process of a pulse starting to appear on the array, i_{mid} has current I_{mid} for the time when all sensor inputs are logic-high, and i_{offset} is the current while a pulse is disappearing from the array. Absolute value is used inside the integral because of the absolute value current circuit, and the absolute value can be split between the various current components because they happen at different times.

An expression is now constructed for the integral of i_{onset} using t_d as the time delay between an edge appearing on two adjacent sensors. The following expres-

sion shows that for the duration of each time t_d, the current flowing from the spatial filter is an absolute value of the sum of all current-DACs with logic-high input sensors. The current-DAC current is the value of its corresponding kernel element multiplied times the current-DAC LSB reference current I_{ref}.

$$\begin{aligned} \int |i_{\text{onset}}|dt &= |I_{\text{ref}} t_d a_0| + |I_{\text{ref}} t_d (a_0 + a_1)| \\ &\quad + |I_{\text{ref}} t_d (a_0 + a_1 + a_2)| + \ldots \end{aligned} \tag{5.9}$$

$$\int |i_{\text{onset}}|dt = I_{\text{ref}} t_d \sum_{n=0}^{8} \left| \sum_{m=0}^{n} a_m \right| \tag{5.10}$$

This equation is true for array activity beginning with sensor 0 or activity beginning with sensor 9, because all of the kernels are symmetric. In fact, these symmetric kernels cause the integral of i_{offset} to be the same as i_{onset}, because i_{offset}integrated over time is the same integrated sum as an onset pulse moving in the opposite direction across the array.

For Equation 5.8 there are now expressions for i_{onset} and i_{offset}. The current of I_{mid} should be equal to 0 with the proper kernel, but in reality it remains as an error term. This yields the final equation for V_{int}.

$$V_{\text{int}} = \frac{2 I_{\text{ref}} t_d}{C_{\text{int}}} \sum_{n=0}^{8} \left| \sum_{m=0}^{n} a_m \right| + [C_{\text{int}} I_{\text{mid}} t_{\text{mid}}] \tag{5.11}$$

Substituting x_0 / v_{array} for t_d, and $V_{\text{err}} = [C_{\text{int}} I_{\text{mid}} t_{\text{mid}}]$,

$$V_{\text{int}} = \frac{x_0}{v_{\text{array}}} \frac{2I_{\text{ref}}}{C_{\text{int}}} \sum_{n=0}^{8} \left| \sum_{m=0}^{n} a_m \right| + V_{\text{err}} \tag{5.12}$$

or equivalently, with $v_{\text{array}} = \omega_t / \omega_x$,

$$V_{\text{int}} = \frac{x_0 \omega_x}{\omega_t} \frac{2I_{\text{ref}}}{C_{\text{int}}} \sum_{n=0}^{8} \left| \sum_{m=0}^{n} a_m \right| + V_{\text{err}} \tag{5.13}$$

Now it has been exactly derived what used to be the arbitrary constant D in Equation 5.5.

Using Equation 5.2 and $v_{\text{array}} = \omega_t / \omega_x$,

$$v_{\text{array}} = \frac{\omega_t}{\omega_x} = \frac{v_s}{\sin \theta_{\text{incident}}} \tag{5.14}$$

Substituting this equation into Equation 5.12 or 5.13, and putting A_{spat} in place of the awkward summations:

$$V_{\text{int}} = \frac{2x_0 I_{\text{ref}} A_{\text{spat}}}{v_s C_{\text{int}}} \sin \theta_{\text{incident}} + V_{\text{err}} \tag{5.15}$$

$$A_{\text{spat}} = \sum_{n=0}^{8} \left| \sum_{m=0}^{n} a_m \right| \tag{5.16}$$

To determine incident angle from voltage output from the STFAP chip, the inverse form of Equation 5.15 it used.

$$\theta_{\text{incident}} = \arcsin \left((V_{\text{int}} - V_{\text{err}}) \frac{v_s C_{\text{int}}}{2x_0 I_{\text{ref}} A_{\text{spat}}} \right) \tag{5.17}$$

Chapter 6

Sonar Bearing Angle Estimation Using Spatiotemporal Frequency Filters: Results

Performance of the STFAP chip was computed from both computer-generated ideal waveforms, and from real-world signals from the microphone array in Fig. 5.5.

Computer-generated waveforms consisted of digital pulses fed directly to the chip, identical to what the preprocessing system would output for perfect incoming array signals. For tests with actual ultrasonic transmission and reception, an ultrasonic beacon was placed 65cm away from the microphone array receiver. The beacon output a continuous tone, and array output was examined for different

angle orientations of the microphone array with respect to the signal path.

Two generations of STFAP were designed and fabricated. Results from both are detailed here. The chief improvement to STFAPv2 over STFAPv1 is much better current matching due to careful circuit design and layout. The linearity and matching of current DACs and current mirroring circuits in general improved greatly between Generation 1 and Generation 2. In addition, the preprocessing functionality implemented in an off-chip FPGA in STFAPv1 was fully integrated into STFAPv2. While the fundamental algorithms for sequencing pulses from square waves are very similar, the integrated preprocessing of STFAPv2 does not use a sampled clock as was necessary in the FPGA to be compatible with its architecture. The logic of the Generation 2 STFAP chip worked satisfactorily, eliminating the extra FPGA and support circuitry previously required by Generation 1. For each stage of the results, both Generation 1 and Generation 2 results will be reported. In addition, Generation 2 of the STFAP improved the design of the postprocessing circuitry which combines the separate digital sign information with the analog magnitude to produce analog signed outputs.

6.1 Preprocessing

6.1.1 STFAPv1

Preprocessing by the zero-crossing detector and FPGA was extremely satisfactory and robust. Over an array angular bearing range of -85° to 85°, the preprocessing functioned with few glitches. Due to the FPGA architecture, operation was necessarily synchronized with a global clock. The 50MHz FPGA clock allowed output pulse edges to be sampled to within 20ns of the edges of the input waveforms. In the future, continuous-time digital circuits will be used, integrated onto the full-custom ASIC. For prototyping ease, however, an FPGA with a very fast clock is a close approximation to the final goal of using continuous-time, asynchronous self-timed circuits for preprocessing. Such circuits will limit activity to only those times in which an input signal edge occurs, greatly reducing power consumption of this block. Additionally, input timing jitter from sampling time-quantization will be eliminated, and signal time resolution will not depend on the speed of a (power-using) global clock.

Output pulse edges were demonstrated to coincide with correlated edges on the inputs. The FPGA code was even able to cope with timing jitter in the input signals large enough to cause one input signal to appear out of sequence with its neighbors. In this case, the output pulse was also out of sequence with its neighboring output pulses, to uphold the relationship of output pulse timing to input

signals, even when they included timing errors. In short, the preprocessing performed transparently, except for the generation of longer pulses.

6.1.2 STFAPv2

In STFAPv2 this preprocessing logic was completely internal to the chip and the FPGA was not used at all. Results of the integrated preprocessing unit matched the FPGA-based performance well. The only divergence in behavior was related to the duration of the created pulses. Both units use counters based on the incoming sensor waveforms to make output pulses equal to a certain number of input carrier wave periods. Whereas the FPGA unit time-sampled the state of the incoming waveforms at a rate equal to about once every 20ns, the integrated logic receives an non-time-sampled binary signal directly from the microphone comparators. Because of this, the edge-triggered counters are prone to glitching if the microphone signal is very noise around the zero-crossing. This results in isolated output pulses that are shorter than they should be, throwing off the entire pulse sequence. In the FPGA case, the sampled nature of the input signals gave a natural debouncing in 20ns chunks of time, and prevented the many of the input glitches from propagating to the counting unit.

One solution which was considered was to use hysteretic comparators on the microphone inputs to the chip. This was considered for Generation 2, but it was decided that there was too much risk in disturbing the precise location of the

zero-crossing threshold to use a hysteretic comparator. Errors in the zero-crossing threshold would translate to timing errors.

Another, possibly safer solution is to use digital one-shot type circuits to prevent the transition of pulses going to the counters until a sufficient debouncing time has passed. Such circuits may minutely affect the timing of the edges, but as long as the timing modification is only routed to the counters this is not a problem. The counters only determine coarse pulse length, and do not affect accurate timing of the onset and offset edges of the pulses, which are directly linked to the edges from the comparators.

6.2 Spatial

6.2.1 STFAPv1

The performance of the spatial filter current-DACs is illustrated in Figures 6.1 and 6.2. These graphs show data from every current-DAC on the chip. The currents in the plots are only positive because shown is the DAC response before application of the sign bit. There are a total of 36 current-DACs in 4 spatial filters. From the graphs of current vs. kernel value, it is apparent that matching is an serious issue. Both inter-DAC matching and linearity suffer from poor matching between transistors. All transistors that were part of the same mirror were in the same local area, and layout techniques such as dummy transistors were used in or-

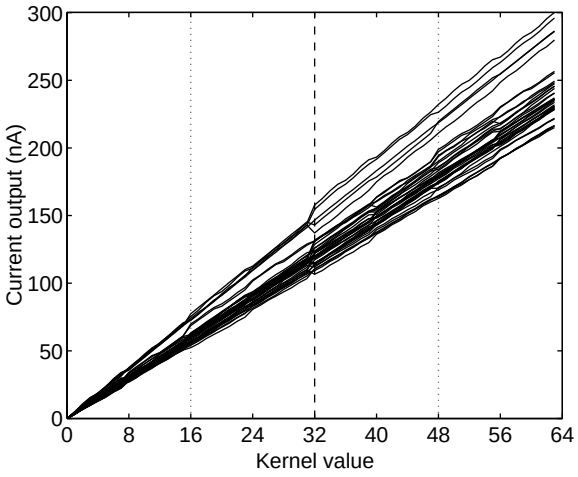


Figure 6.1: Current output of all current DACs on chip for positive kernels, 1^{st} Generation STFAP.

der to make the electrical and physical environment of each transistor similar to its counterparts. However, these efforts were not enough. From the experience with these first generation chips, extreme attention to layout, and the use of common-centroid layout techniques were clearly necessary to improve current-DAC linearity and matching.

Statistics from the whole chip show that linearity is passable, with an overall RMS error of less than 3nA from a linear fit of each curve. As is expected, most non-linearities happen at major bit boundaries. Inter-DAC matching for each kernel coefficient stays in the range of 10–12%, measured by the standard deviation divided by the mean at each current value. This mediocre current-DAC matching is also representative of inter-kernel-element matching, as well as inter-filter matching.

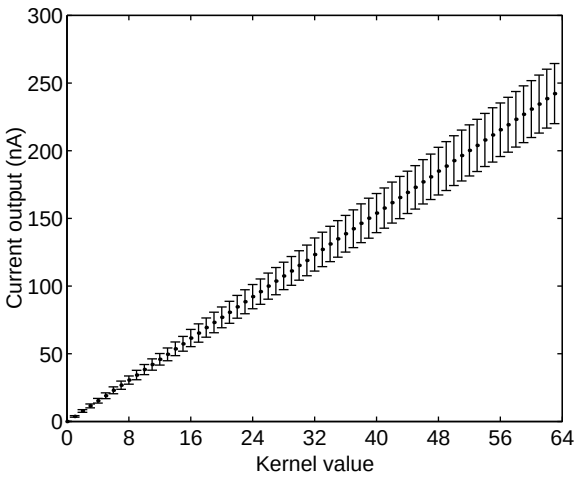


Figure 6.2: Collective statistics (mean $\pm$ std. deviation) of all kernel DACs on chip, 1^{st} Generation STFAP.

It is worth noting that ultimately exact matching is not important, because kernels are programmable and set at runtime. Characterization of each current-DAC allows fine-tuning its current output by simply changing the number written to it during initialization. Instead of the ideal kernel value, a value is written matching the actual current output desired. In normal operation, kernel values are only written once, so this is not an undue burden. Also, all current-DACs only need be characterized once to determine process-induced mismatch.

Other factors affecting performance of the spatial filter subsystem are inherent current DC offsets of the filter outputs and the performance of the absolute value circuits. Mismatch again caused a wide range of DC offset currents from the filters. While ideally there should be no inherent DC current from a spatial filter, all of the filters exhibited different offsets due to mismatch. It is difficult to separately

analyze the absolute value circuits and the sign circuits in the current-DACs. The combination of the current-DAC sign circuit and the absolute value circuit produced currents within 10% of the positive current-DAC values.

From Figure 6.1, it is apparent that a few traces are on a separate branch from the majority of the DAC response curves. Most of these errant plot-lines are from the same filter. The other three filters on chip exhibited better matching, and these three were used for the rest of the tests. Matching and linearity statistics for this filter were similar to statistics for the chip as a whole. In the rest of this section data from this filter will be presented.

6.2.2 STFAPv2

The performance of the spatial filter current-DACs is illustrated in Figures 6.3 and 6.4. In the second generation chip, due to very careful circuit design and layout, matching increased by almost a factor of 2. Inter-DAC matching for each kernel coefficient increased to 5-6% for most of the kernels, again measured by the standard deviation divided by the mean at each current value. The graphs show a much tighter clustering of current traces. This was the norm for all chips, and all chips performed well. In contrast, some chips from STFAPv1 were actually unusable because mismatch caused some non-robust circuits to be completely outside of their usable operating range.

An additional feature of the second generation chips was a separate trimming

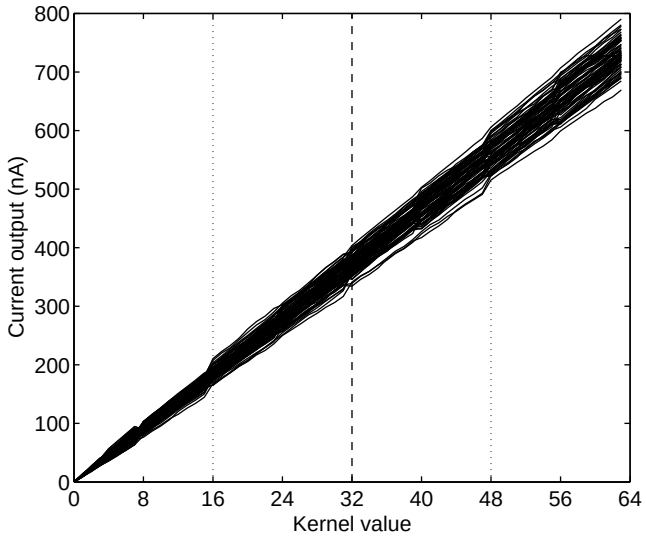


Figure 6.3: Current output of all current DACs on chip for positive kernels, 2^{nd} Generation STFAP.

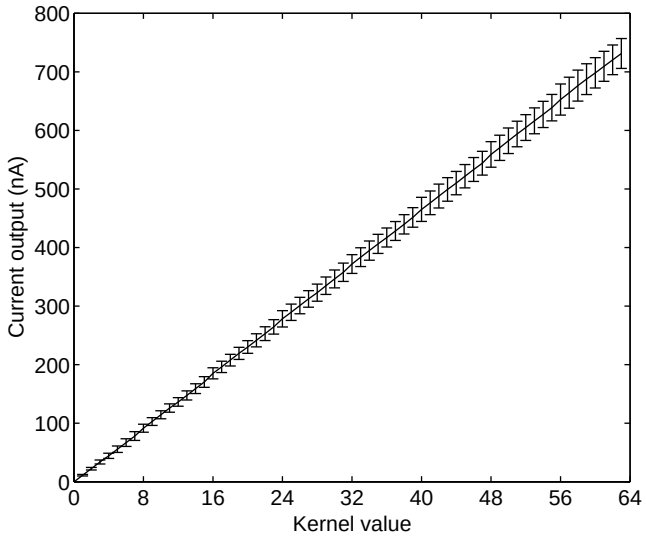


Figure 6.4: Collective statistics (mean $\pm$ std. deviation) of all kernel DACs on chip, 2^{nd} Generation STFAP.

current DAC on the output of the each kernel element current DAC. With this trimming circuit, any DC offsets could be completely eliminated. However the matching of most kernel DACs in this generation of chip had improved so much, that trimming more than 1LSB of the kernel magnitude was never necessary.

6.3 Temporal

Integration of the values proceeded correctly. The only minor problem was that, due to layout imperfections in the Generation 1 STFAP chip, the ratios of the various sizes of integration capacitors did not match the intended ratios. Most likely, extra parasitic capacitances added to explicit Poly1-Poly2 capacitances and affected the actual final capacitive ratio. This phenomenon is discussed further in Section 6.6.

As will be shown below in Section 6.6.1 the capacitance matching for Generation 2 was able to be characterized much more explicitly due to improvements in circuit design and experimental setup. The deduced capacitance values from the data in that section show capacitive ratios mostly consistent with designed values, with slight deviation from design value for the 1.5pF capacitor. This could be because it is the smallest value, involving the fewest constituent sub-capacitor units, and thus more prone to error. Or it could simply be because the experimental curves for this smallest capacitance involve the fastest signals and are the most

difficult data points to capture. The number of data points available for analysis for the 1.5pF integrating capacitance was much less than the other capacitances, which could also lead to errors in characterization.

6.4 Sign detection

The sign-detect circuit functioned satisfactorily in both STFAPv1 and STFAPv2. For STFAPv1, sign output was always correct so far as it was able to be measured. The angle at which sign changed from plus to minus was as close to 0° as could be determined with the experimental setup.

Because of the time resolution of the experimental setup for generating synthetic waveforms increased five-fold between STFAPv1 and STFAPv2, the second generation chip was able to be more precisely studied. Problems were found that although the sign-detection circuit functioned properly in most instances, at near-normal (near 0°) angles. Where the sign changes, wavefront velocity across the array is extremely fast. It appears that the kernel-based sign-detection circuit responds too slowly, and gives erroneous results very near 0°. For an example of this behavior, see Figure .

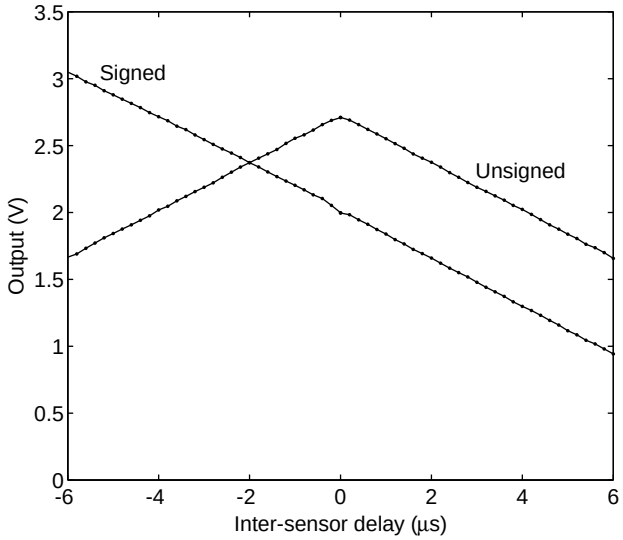


Figure 6.5: Illustration of Post-processing unit correctly converting unsigned analog magnitude voltage to signed analog voltage, incorporating sign-bit generated on chip. Results from 2nd Generation STFAP.

6.5 Postprocessing

Because of improvements to the circuit design, testing of the postprocessing unit were successful in STFAPv2 where they were generally unsatisfactory in STFAPv1. For an example circuitry to combine digital sign-bit with analog magnitude, see Figure 6.5. The unsigned curve has been generated by passing actual synthesized digital waveforms to the chip microphone inputs. (See Section 6.6.1)

6.6 Entire system performance

6.6.1 Computer-Generated Input Test Signals

The first test of STFAP system performance was conducted using ideal digital signals supplied to the chip from a computer. The waveforms were long digital pulses simulating the output of preprocessing. Different array wavefront velocities were simulated by the synthesized signals. This test was a baseline characterization of the processing circuitry on chip.

6.6.1.1 STFAPv1

The first tests of the synthetic waveforms was performed on the first generation of STFAP chip. This first version of the chip suffered from the matching problems illustrated earlier in Section 6.2. The best spatial filter was used for the data below.

Inter-sensor delay was swept between -35μs and 35μs, which for my microphone array would correspond to array speeds higher than 94m/s. All four spatial filters mentioned above were tested, as were four separate integrating capacitances available in the system. Due to the relatively slow speed of the pulse-generation hardware used to test this chip, time resolution of less than 1.3μs was not possible. The limited the fastest array speed to 2538m/s. For these slow wavefronts, the smallest values of C_{int} had too few data points in the usable voltage range to effectively analyze. The largest integrating capacitance, 12pF, yielded the clearest data.

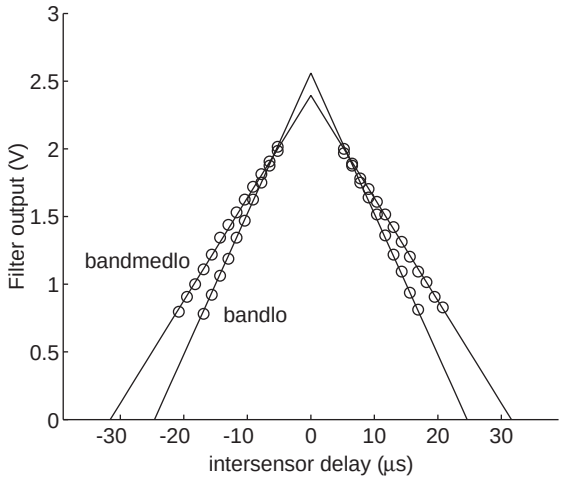


Figure 6.6: Computer-generated inputs and C_{int}=12pF. Kernels: bandlo, bandmedlo. Chip: 1^{st} Generation STFAP.

Plots of this data for each of the four spatial filter kernels can be seen in Figures 6.6 and 6.7. In the STFAP system, integration begins at 3.3V, so the salient V_{int} value at any point is measured from that reference voltage. Because voltage outputs of the filters will be inversely proportional to velocity, these plots use inter-sensor time-delay so that a linear and proportional relationship can be observed.

From the data plotted, slope and y-intercept parameters are extracted. These parameters represent $2I_{\text{ref}}A_{\text{spat}}/C_{\text{int}}$and V_{err} respectively from Equation 5.11. The theoretical and experimental results for these parameters can be seen in Table 6.1. As you can see from the values, the experiments match well with theory: the experimental data for the curve slopes differ by 0–5.5% from the calculated parameters. The most probable sources of error are the matching of I_{ref} currents and C_{int} capacitors. As for V_{err} values—these stem from non-zero current from the filters when

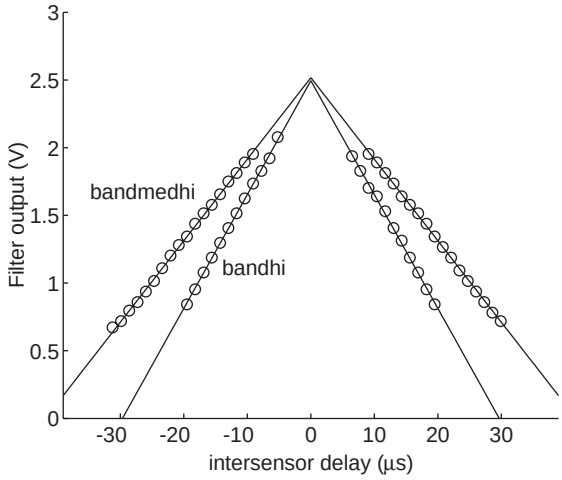


Figure 6.7: Computer-generated inputs and C_{int}=12pF. Kernels: bandmedhi, bandhi. Chip: 1^{st} Generation STFAP.

	Theoretical		**Experimental**	
Filter name	$2I_{\text{ref}}A_{\text{spat}}/C_{\text{int}}$	V_{err}	$2I_{\text{ref}}A_{\text{spat}}/C_{\text{int}}$	V_{err}
bandlo	110V/ms	0V	104V/ms	0.76V
bandmedlo	77.6V/ms	0V	75.8V/ms	0.91V
bandmedhi	60.0V/ms	0V	60.2V/ms	0.787V
bandhi	83.1V/ms	0V	84.1V/ms	0.81V

Table 6.1: Theoretical and experimental values for spatiotemporal filter parameters. Data from 12pF integrating capacitor of STFAPv1.

all sensor inputs are logic-high. This occurred because I didn't calibrate the spatial filter kernels from current data, and used kernels whose sums were close (less than 2) but not always equal to zero. The V_{err} values show up on the graphs as an offset from 3.3V. With calibration, V_{err} can reduce to nearly zero, and thus give a larger signal range. In addition, although C_{int} values were designed to increase in powers of two, the experimental data shows that this is not exactly the case, the real multiples being 1.6 or 1.8 depending on the size of capacitors being compared. It is more difficult to extract accurate slopes from this experimental data, because there are fewer points in the high-slope curves before the signal saturates. But it is likely that extra parasitic capacitances not carefully considered are affecting the ratios, as mentioned in Section 6.3.

6.6.1.2 STFAPv2

From the improved current matching of the second generation of STFAP chip, all filters behaved well. One of the spatial filters from this design is illustrated in the data below.

For the testing of the second generation chip, an improved experimental setup allowed 200ns time resolution on the synthesized signals. Instead of being generated by a Digital I/O card in a PC, a fast PIC microcontroller from Microchip was used to create the pulse signals. Data for the different spatial filters already showed a good match for theoretical expectations even with the less precise generation 1

chips. Also, with the increased time-precision available in the experimental setup to test generation 2, data is now able to measured for all capacitor values. Because of these two factors, for this version of the chip data from one spatial filter is displayed, "bandhi", integrated with 4 separate capacitors values. The data measured also matches acceptably with theory.

The most striking difference between chip generations can be seen in Figure 6.8. Dense data points for all capacitance values are displayed. With the increased time resolution of the input waveforms, the exceptional linear response of the filters can be seen. In addition, the new circuitry of STFAPv2 produces meaningful data points covering the middle of the graph, corresponding to the region around 0μs delay. This region was missing from STFAPv1, because it was completely out of range. The datapoints in Figure 6.8 which are obviously saturated are all located below 0.8V, making for easy discrimination of useful data from errant points.

6.6.2 Microphone Signals from Ultrasonic Beacon

With baseline performance now firmly in hand, the practical results of the full system in action is now presented. With a stationary ultrasonic pinger and a rotating stage for the microphone array, I tested the reception of real sonar pulses into the system.

(for filter bandhi)	**Theoretical**		**Experimental**	
Integrating Capacitor	$2I_{\mathrm{ref}}A_{\mathrm{spat}}/C_{\mathrm{int}}$	V_{err}	$2I_{\mathrm{ref}}A_{\mathrm{spat}}/C_{\mathrm{int}}$	V_{err}
1.5pF	1500V/ms	0V	1240V/ms	0.93V
3pF	750V/ms	0V	680V/ms	0.74V
6pF	375V/ms	0V	346V/ms	0.61V
12pF	188V/ms	0V	174V/ms	0.61V

Table 6.2: Theoretical and experimental values for spatiotemporal filter parameters. Data from bandhi and all integrating capacitors of STFAPv2.

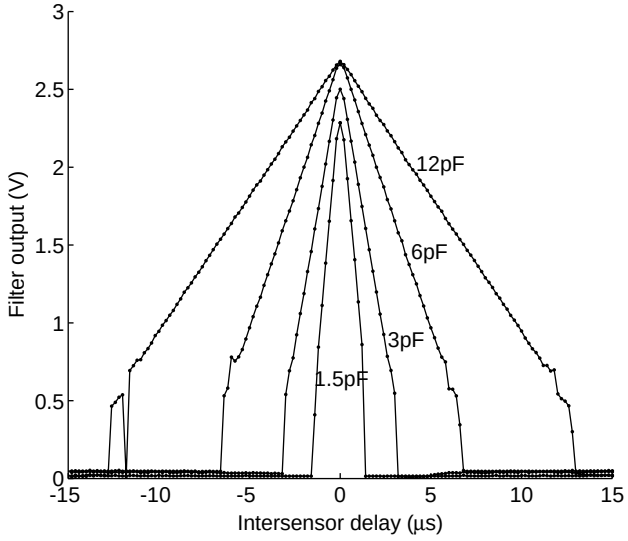


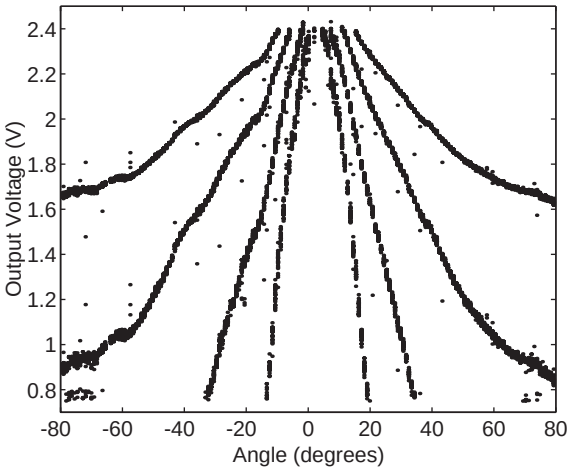
Figure 6.8: Computer-generated inputs and kernel bandhi, for different values of C_{int}. Chip: 2^{nd} Generation STFAP.

6.6.2.1 STFAPv1

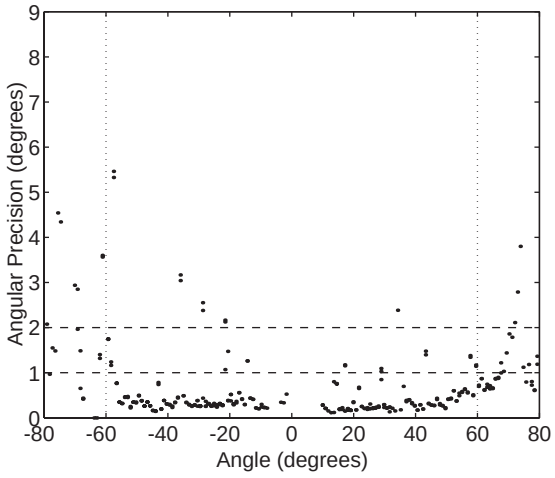
The four spatial filters' responses can bee seen in Figures 6.9(a.), 6.10(a.), 6.11(a.), and 6.12(a.).

It is important to remember that these plots are plotted against bearing angle, not inter-sensor delay as are Figures 6.6 and 6.7. As such, they will include the effect of the sine function and not be purely linear. However, they will be approximately linear near 0° because the sine function is approximately linear there. Also, I have limited the voltage excursion of the STFAPv1 system so that voltage buffers in the chip stay in their functional range. This leaves a gap around 0° bearing for some curves. The steepest curves on both sides of 0° result from the 1.5pF integrating capacitance, and the shallowest curves from 12pF integrating capacitance.

Visible on some curves is a slight waviness for negative angles. This is probably due to a wall on the negative-angle side of the experimental setup. Care was taken to reduce reflections by placing the testing apparatus near the middle of the room, and by acoustically damping the wall surface. I did not have a true anechoic chamber in these experiments however, and the data shows the effect of sonic contribution from this wall. Interference was also unavoidable because these experiments used a continuous tone instead of short bursts. Had bursts been used, the first returns would have been accurate, coming from the shortest path from beacon to array, and the longer path from beacon to wall to array could have been ignored. The continuous-tone experiments that were actually implemented meant

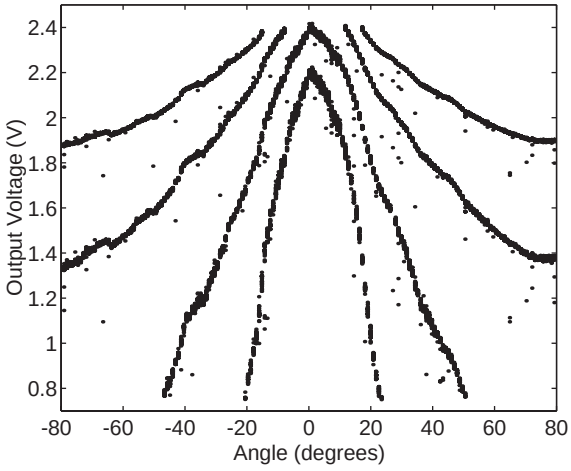


(a)

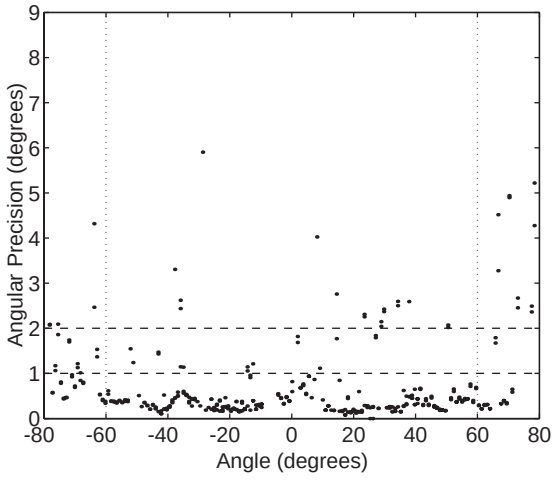


(b)

Figure 6.9: STFAPv1 system responses with the **bandlo** spatial filter to real ultrasonic beacon inputs. (a) Raw voltage outputs. (b) System precision measure at each angle.

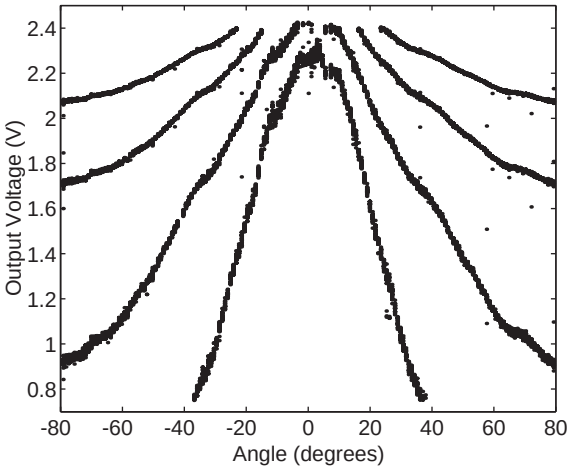


(a)

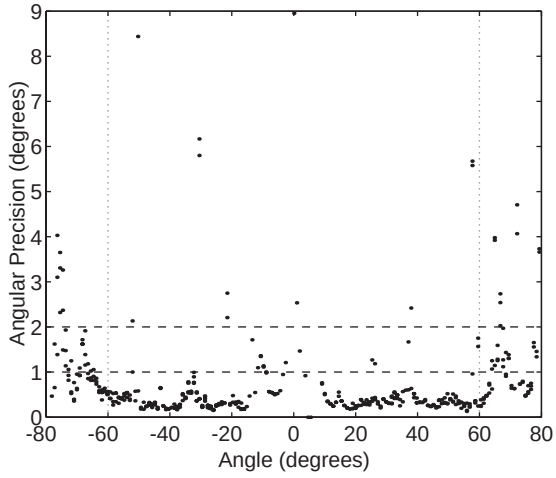


(b)

Figure 6.10: STFAPv1 system responses with the **bandmedlo** spatial filter to real ultrasonic beacon inputs. (a) Raw voltage outputs. (b) System precision measure at each angle.

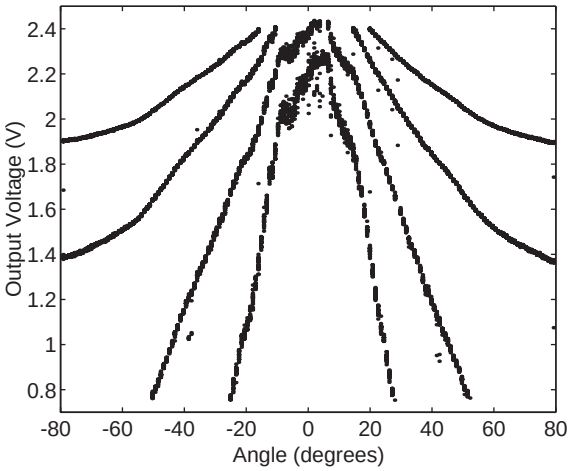


(a)

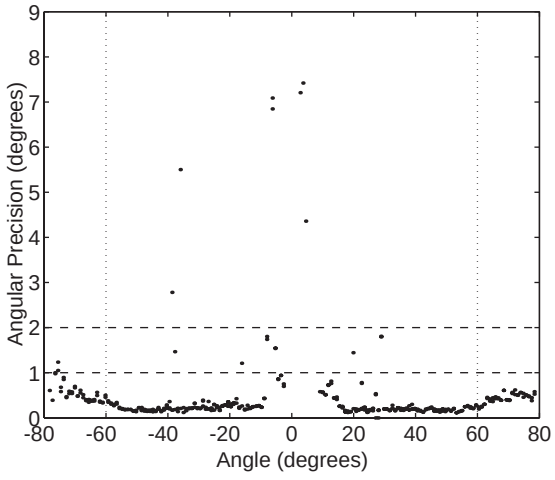


(b)

Figure 6.11: STFAPv1 system responses with the **bandmedhi** spatial filter to real ultrasonic beacon inputs. (a) Raw voltage outputs. (b) System precision measure at each angle.



(a)



(b)

Figure 6.12: STFAPv1 system responses with the **bandhi** spatial filter to real ultrasonic beacon inputs. (a) Raw voltage outputs. (b) System precision measure at each angle.

that at every instant of time the signals received by the array were a superposition of direct transmissions and faint wall-bounced signals. Despite this experimental drawback, the data remains remarkably clear.

These raw data plots contain points for 20 observations at the each angle, allowing us to compute statistics. Visual inspection of the data shows that there is not much vertical spread of points at the same angle, a qualitative figure of merit. A useful quantitative figure of merit is precision at a given angle, arrived at by dividing the standard deviation at an angle by the slope of the mean at that angle. By this measure, the smallest angular difference that can be reliably distinguished from this data can be shown. Plots of precision for all values of C_{int} for a given spatial filter kernel are shown in Figures 6.9(b.), 6.10(b.), 6.11(b.), and 6.12(b.). In each graph, a maximum of 2 outlier points have values that place them off the top of each graph in the usable range from -60° to 60°. The exception to this is the obviously noisy center part of 2 curves in the plot for the bandhi spatial filter, and 1 curve in the plot of the bandmedhi spatial filter. The precision in these center regions is too poor to be included on the precision graphs. This data could be excluded by changing the voltage thresholds for acceptable data on-chip, but was included here to show more complete data for the chip. The majority of points, however, are within a precision of 1°, and most of the few remaining points still show better than 2° precision.

6.6.2.2 STFAPv2

Improvements in the experimental setup occurred for the measuring of STFAPv2. The most important improvement is the use of pulses instead of continuous tones from the beacon. This should eliminate superposition of extraneous echos from polluting the beacon signal. There is still some waviness in the real response of the array to bearing angle, but it is possible this is the result from nonidealities of the physical transducer array. In addition the response is markedly more symmetric, probably due to reduced effects of echo interference from nearby objects.

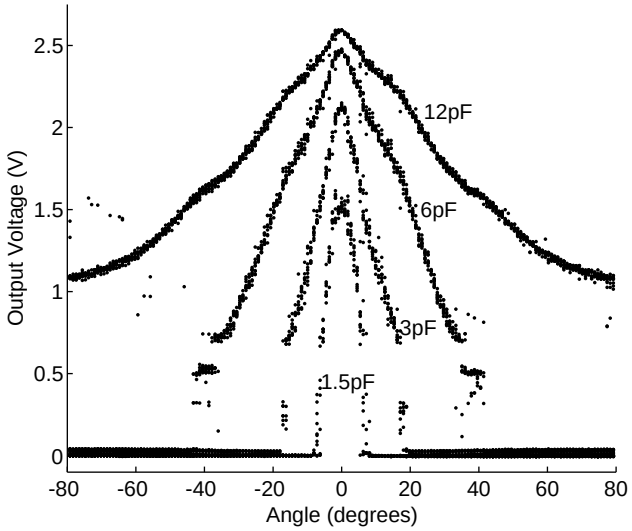
From Figure 6.13(a.), some improvements can be seen in the response of the 2^{nd} Generation chip to real ultrasonic beacon pulses. The major improvement is the complete continuity of response through the 0° angle. In the previous version of the chip, internal circuitry was out of range for these points, making it impossible to read bearings from this critical region at the center of the angular range of the array. Now however, full readings are available. This increased information is reflected also in the graph of precision in Figure 6.13(b.). Instead of the gap seen in previous precision graphs for STFAPv1 where precision and data were undefined, here there is a dense amount of datapoints from all 4 filters squarely in the region of 1° precision. Besides this region near 0° , overall precision is similar in this generation of the chip to the previous generation: at many angles precision is computed to be better than 1°, and better than 2° precise for almost all of the rest within the range $(-60^\circ, 60^\circ)$. While the circuits of STFAPv2 became more

precise, it appears that external factors such as microphone amplification and filtering, microphone quality, and the propagation factors of ultrasound itself, may still contribute enough errors to rule out a major increase in overall bearing precision. Major over improvement remains, however, by the expansion of the usable angular range of the array to angles near 0°.

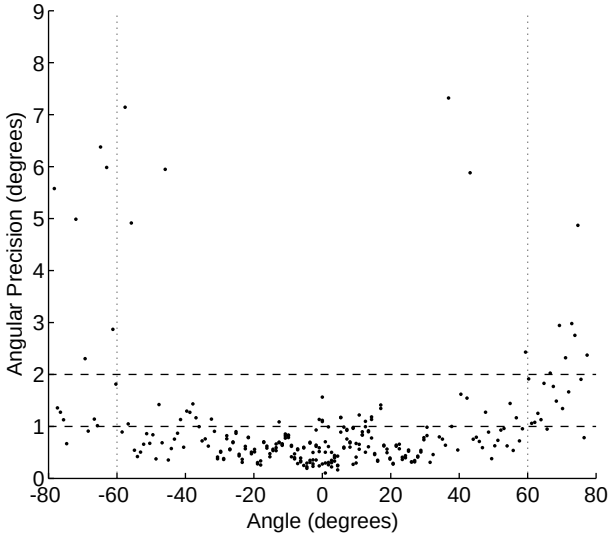
6.6.3 STFAP Accuracy

The most important figure of merit for the STFAP system is accuracy. Accuracy measures how the processor determines the absolute angle value, and not merely how well it distinguishes two angles from each other. To test angular accuracy, I first made plots for each of the 16 filter outputs (4 spatial filters, with 4 temporal integrators each.) I then fit these data plots to the theoretical formula, Equation 5.17. Two parameters for each filter were extracted from these fits: V_{err}, and a parameter m encompassing $v_s C_{\mathrm{int}}/(2x_0 I_{\mathrm{ref}} A_{\mathrm{spat}})$. For complete system operation, the information from all 16 filter outputs are used together. Since the arcsine function of Equation 5.17 is non-linear, instead of averaging the separate computed $\theta_{\mathrm{incident}}$ expressions together, the separate solutions to the linear expression inside the arcsine function were averaged. The arcsine function is then applied to this average to arrive at the final $\theta_{\mathrm{incident}}$ estimate, as shown in the Equation 6.1.

$$\theta_{\mathrm{incident}} = \arcsin\left(\frac{\sum_{i=1}^{16}\left(V_{\mathrm{int}i} - V_{\mathrm{err}i}\right) m_i}{16}\right) \tag{6.1}$$



(a.)



(b.)

Figure 6.13: STFAPv2 System responses with the **bandhi** spatial filter to real ultrasonic beacon inputs. (a) Raw voltage outputs. (b) System precision measure at each angle.

6.6.3.1 STFAPv1

Figure 6.14 shows the angle estimates that result from the measured data. Apparent is a loss of accuracy around 0°, which mirrors the lack of reliable data near 0° in the raw data and precision graphs from each of the filter outputs in Figures 6.9, 6.10, 6.11, and 6.12. Since the graph is monotonic, these problematic points can be easily identified in the raw data from the voltage range they occupy. They are separate from and are in no danger of being confused with the more reliable data from other angles. It may be possible to fit the data from angles near the center of the range better by fitting the curve with a more generalized function. This may also help overall accuracy for the entire angular range. In the error plot in Figure 6.15, the errors are not distributed randomly, but have definite trends. Improving the method of fitting would eliminate these more deterministic deviations from the true angle. Despite these imperfections, overall accuracy, measured by the standard deviation of the error from ideal, was still 1.4° for the entire range of angles between -60° and 60°. This figure improved to 1.0° error if angles between -3° and 3° are omitted from error calculations.

6.6.3.2 STFAPv2

Again, in Figure 6.16, the predominant improvement of the 2^{nd} Generation STFAP is the successful processing of angles near normal to the array. The same simple linear fit procedure was used to produce coefficients to translate each sonar

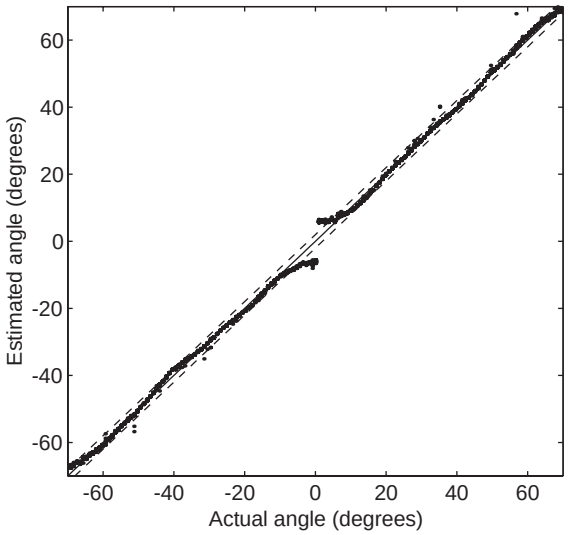


Figure 6.14: Graph of estimated angle vs. real bearing angle. Dashed lines show $\pm 2^\circ$ from the ideal.

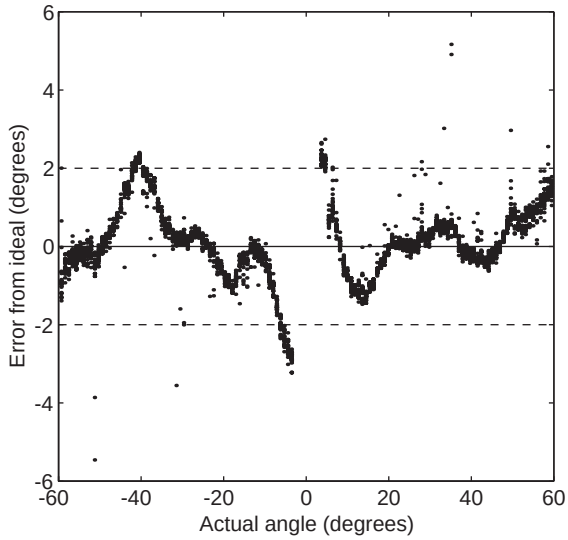


Figure 6.15: Error plot of the difference between estimated angle and true angle. Dashed lines show $\pm 2^\circ$ from the ideal.

bearing voltage output to a corresponding angle. Instead of the unfortunate discontinuity in the center of Figure 6.14 for Version 1 of the chip, Figure 6.16 shows that Version 2 produces a continuous line of estimates with little spurious data on the whole interval $(-70, 70°)$.

Figure 6.17 shows the errors. Again, due to successful circuits the center of the graph is filled in with actual points of very respectable accuracy. As before, overall the graph of error shows some structure and is not completely random. The points are a little more distributed for the Version 2 chip results seen here, but there is a trend which possibly could be eliminated with a higher-order voltage-to-frequency mathematical model. Accuracy is still very healthy, and overall for the range of $(-60, 60°)$ has significantly improved to a standard deviation error of $1.2°$, this time including the correctly processed region around $0°$.

6.6.4 Sonar Signals Reflected from Real Objects

To finally test the system in a real use of reflected sonar, an ultrasonic pinger and the receiver array were placed together, viewing a scene with two tall cylindrical targets placed on a bench top. No Temporal Gain Compensation (increasing gain over time to boost distant pulses) was used. The setup was almost exactly the same as for the beacon tests, except that both sonar transmitter and array receiver were both directed at the scene. STFAP version 1 chip was used for the processing.

Figure 6.18 shows position of sonar transmitter/array receiver (at [0,0]) and

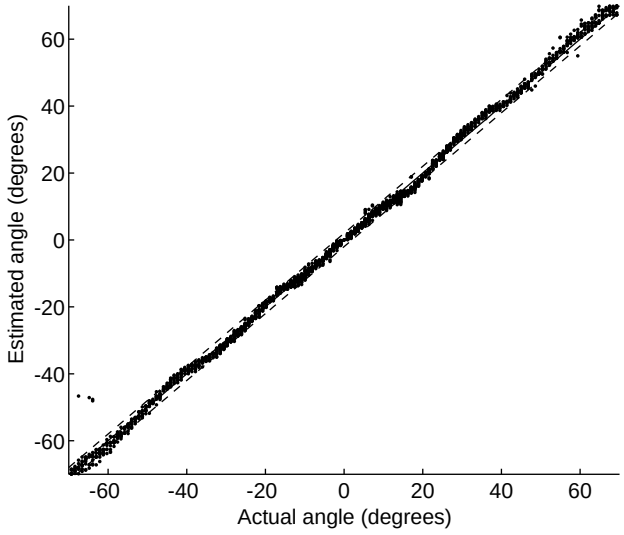


Figure 6.16: Graph of estimated angle vs. real bearing angle. Dashed lines show $\pm 2^\circ$ from the ideal.

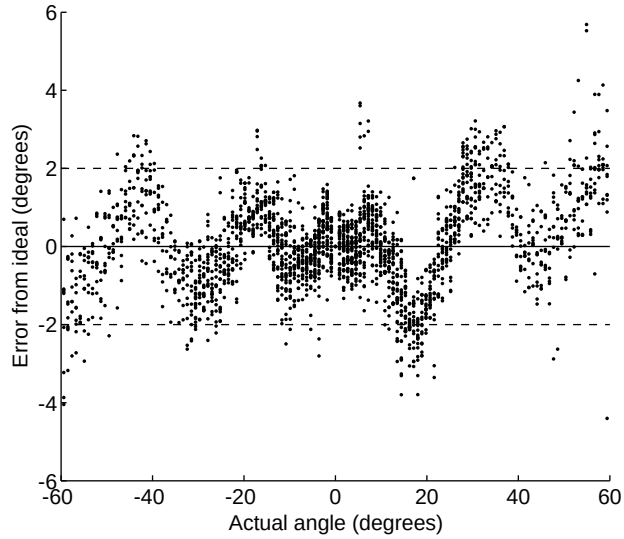


Figure 6.17: Error plot of the difference between estimated angle and true angle. Dashed lines show $\pm 2^\circ$ from the ideal.

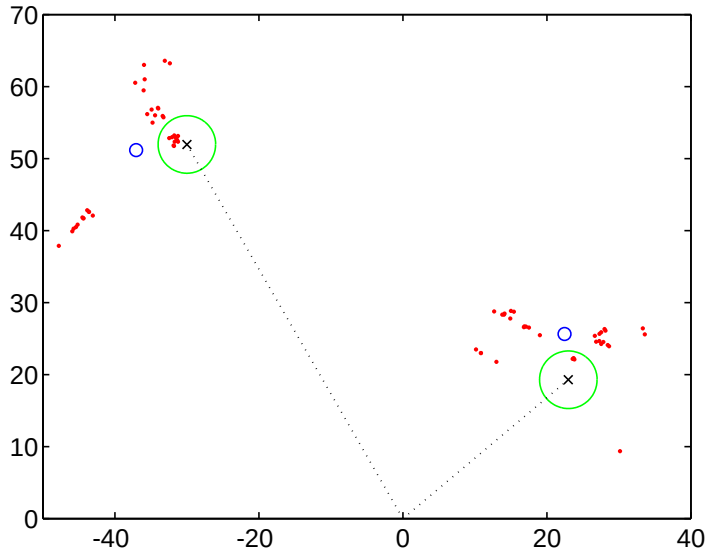


Figure 6.18: STFAPv1 computing the 2-D position of two objects in a full sonar trial.

position of the two cylindrical targets (circles). In addition, dots corresponding to 2-D position estimates from array range and angle data are also plotted. The locations of the dots are computed using angles computed by the linear fit method described in Section 6.6.3. Range was computed from computed time delay from ultrasonic transmission to reception and an assumed value for the speed of sound in air of 340m/s. Sound speed is known to be variable due to changes in pressure and temperature, but to a first order this figure should be correct.

Results of this trial are encouraging. The most noticeable systematic error is the "shadow" cast by the object in the upper-left area of Figure 6.18. This is due to the long pulse used in our experiments. Because our amplification was simple, and Temporal Gain Compensation (TGC) was not used, it was necessary to

increase the length of the pulse to ensure adequate reception. A longer pulse allows both the sonar transmitter and our array microphones to "ring up" or react to the abrupt beginning of the carrier pulse. The side-effect of a longer pulse, however, is that multiple bearing readings can be taken along the length of the pulse. The STFAP will attempt to read the bearing from a sonar pulse as soon as it has finished reading the bearing of another. If the same pulse is present at the array, it will readout the same bearing again, at a later time. This translates to multiple readings, with the same bearing but different apparent distances. This behavior explains the additional clusters behind the upper-left object.

Other errors are present in this test that are random errors of measurement, showing no systematic structure. It is expected that these random errors can be lessened with more careful amplification and signal conditioning of the array microphone inputs, including TGC. In addition, better amplification circuitry would allow shorter ultrasonic pulses to be used, eliminating multiple readings and increasing depth precision for all readings.

6.7 Summary

The processing described enables a sonar array to extract not only range information, but bearing angle as well. The implementation is smaller in size and power than comparable solutions. Because it is efficient for continuous-time ana-

log circuits to preserve phase information of the incoming waveforms, more sophisticated processing than ensemble ranging is possible: actual phased array processing. Instead of examining pulse onset and offset, phase information at every time of a received waveform can be examined. This synchronous processing allows multiple returns from multiple angles to be correctly analyzed, even if they are so close together in time that they appear to be one long pulse.

With very simple low power circuits, 1° precision was obtained in many instances from basic experiments receiving an actual ultrasonic pulse. Accuracy was tested by fitting the data to the theoretical equation for system response. Overall accuracy of angle estimates was found to have a standard deviation of 1.2° for the entire range from -60° to 60°. Full 2-D sonar operation was also demonstrated. A careful study of the effects of off-chip factors in the sonar system is necessary to increase accuracy.

By combining this compact implementation with compact MEMS sensors, the ultimate goal of a small rich-data sonar sensor package yielding both range and angle information can be obtained. It promises a practical means to sense objects and the environment in three dimensions for mobile or resource-limited applications.

Process	0.5μm 3M2P CMOS
Chip size	3.0mm $\times$ 1.5mm
Power Consumption	1.02mW
Ultrasonic frequency	40kHz
Number of microphone inputs	9
Baseline (Widest microphone separation)	26.5mm
Precision (Standard deviation/slope of mean)	1.0° (-60° to 60°)
Accuracy (Standard deviation, estimated-actual angles)	1.2° (-60° to 60°)

Table 6.3: Chip and sensor summary.

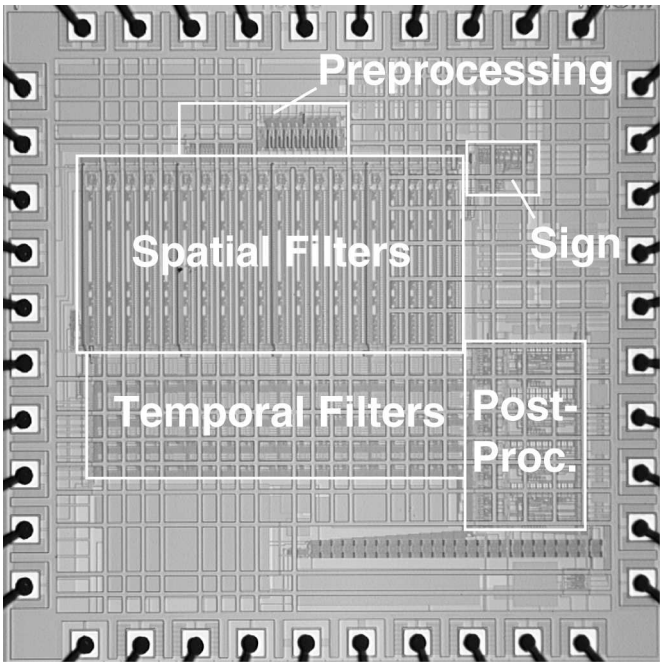


Figure 6.19: STFAPv1 Chip micrograph.

Chapter 7

Sonar Bearing Angle Estimation Using Spatiotemporal Frequency Filters: Applications

7.1 Mobile Robots

For decades, research with autonomous mobile robots has approached the ideal of a economical robust 3-D sensor. A sensor able to detect in three dimensions is essential to enable navigation and maneuvering around static and moving objects. Unfortunately, currently there are no sensors that fit both the robust and economical demands. Laser-rangefinding is potentially very accurate, but is usually expensive in terms of money and computation, because of the precision required

for measurements. Traditional sonar using wide-angle sensors and Time-Of-Flight (TOF) processing is very economical, but not very robust. Still, many groups concentrate on making TOF sonar systems viable because their low cost is extremely attractive compared to other approaches.

There are some well-known fundamental problems with sensing the world with regular sonar sensors that are typically wide-angle, and using TOF to determine depth only. The major problem is that with the low frequency used for sonar in air (typically near 40kHz), almost every surface looks smooth compared to the wavelength of the sound waves. Hence every surface is specular, and reflects sound like a mirror instead of presenting a diffuse reflection in all directions. This means that sonar returns typically only come back from ultrasonic energy directed normal to surfaces, and sound transmissions directed at a surface on an angle will merely be bounced off in another direction and not back to the receiver unit. In other cases, transmissions will return to the receiving unit after multiple ricochets. If the receiver assumes the return comes from a simple bounce off of one object, the results will be erroneous. The second fundamental problem is that sonar sensors typically are very wide angle. To build a very focused transmitter or receiver of low-ultrasonic frequency sound would require a very large physical apparatus. Instead most sonar transceivers, to keep a moderate size, have a very wide-angle field-of-view. This allows reflections to come to the receiver from a very large solid angle. Because it is not possible to determine the precise return angle of sonar re-

turns to these sensors, the ambiguity of the receiving system is increased. The return could be coming from straight ahead, or from an object off to the side. Because TOF sensors only receive one real-valued datapoint per output pulse, the distance of the imaged object from the sensor, this angular ambiguity translates to a fundamental ambiguity in the measurement of the 3-D environment.

7.1.1 Prior Work

To remedy this problem, many processing approaches have been developed to take the ambiguous data from multiple TOF sonar sensors, combine the data using heuristics, statistics, neural networks, or other methods, and deduce the proper 3-D information from the collection of uncertain data. This type of processing can be quite intense and complex, needing hundreds to thousands of MIPS to get close to real-time operation. To implement a processing subsystem of the appropriate computation complexity to refine TOF sonar data requires significant expense, both in terms of monetary cost and engineering costs such as power and system complexity. Considering that the prime motivation behind using sonar as a sensor modality is its relative economy, these approaches tend to be counter-productive. They are analogous to having an unfocused camera and trying to use advanced image processing in order to get a clear image. It can possibly be done, but compared to starting with a camera that is properly focused it's a waste of resources.

The typical approach used with conventional TOF sonar sensors uses Polaroid

sonar transducers with processing on a PC or DSP. Often a ring of sensors is used, with each sensor viewing a sector of space looking out from the perimeter of the robot. While receiving, each Polaroid sensor uses 0.5W of power. Thus for a ring of 24 sensors, 12W are being consumed just in the receive circuitry. The extra processing necessary to refine the multiple TOF range readings has been accomplished by different groups in different ways. For software flexibility, standard Personal Computer (PC) hardware has often been used, drawing 100W or more.

The most basic naive approach to modeling a sonar return to a wide-angle sensor is to assume that the pulse is being received from directly ahead. Called the Centerline Model, this approach is acknowledged as naive but may be used for efficient computation to get a rough picture of objects in the proximity. In reality, given a range data point from a sonar sensor, the return could be from anywhere in an arc equidistant from the receiver within the angle of view. For the Polaroid 6500 sensor, this angle of view is approximately 120°. Some have modeled the uncertainty along this arc in a Gaussian fashion, assuming a higher likelihood that a pulse comes from straight ahead [72]. In the reference, Elfes combines this probability distribution with an "occupancy grid", mapping the entire room and updating probabilities of objects being present at each point of the grid. Note that this method *requires* internal mapping functionality to work, along with accurate self-knowledge of robot movement in addition to sonar.

The motivation for a Gaussian probability distribution probably is due to its

passing resemblance to the sensitivity curve of the Polaroid sensor. However, close objects' return energy can easily be greater than the receive threshold and be sensed even if they are positioned on a low sensitivity bearing on the edge of the possibility arc. For these objects, it is arguable whether or not the Gaussian distribution is valid. Recent work uses a uniform possibility for an object's position along the equidistant arc. Choset et al. [73] use what they call the Arc-Transversal Median (ATM) algorithm with uniform-probability arcs. From each sonar return they create an arc in their map. Intersections of arcs meeting certain criteria are assumed to be a real location of an object. The processing for the ATM algorithm is claimed to be modest: "This approach works entirely in software in real time, so no additional hardware modifications have to be made or purchased." However, this implies operation on a PC, with its associated 100W or more of power consumption. It is unknown just how much processing power is actually necessary for the algorithm to succeed or what kind of processing hardware would be necessary in the absence of a PC. In addition to the unknown cost of computation, there are other practical issues for the algorithm's successful operation. To make a sufficiently detailed map of the surroundings, returns of the same object or surface must be taken from multiple positions. This necessitates movement of the robot to scan its surroundings for an accurate picture. It also requires knowledge of its own relative positions as it moves, and the ability to keep track of a map. Multiple range samples are fundamentally necessary for the processor or "planner" to com-

pensate for the uncertainty of each TOF sonar return. As the Choset et al. note, "a sonar sensor only provides a real number. This is all the planner knows."

To increase performance, specialized processing using embedded DSPs for dedicated processing has been demonstrated. In [74] Fazli and Kleeman use an optimized DSP system to extract 3-D data from sonar scans in near real-time (15scans/s). They use the traditional "ring" of sonar sensors around the periphery of a robot. It is quite an elaborate system, employing 24 sets of transmitter/receiver transducer pairs, 6 "slave" DSP boards, and one "master" DSP board to coordinate the slaves. The transducers are standard Polaroid sonar transmitters and receivers. Each slave board uses one Analog Devices ADSP-2189M DSP and two Texas Instruments ADS7862 ADCs. To process the data from all of the slave boards, a master board with another ADSP-2189M DSP controls everything, and computes bearings using triangulation. To operate the system, all 24 transmitters emit a uniquely-coded sonar pulse simultaneously, and the 24 receivers note when they receive each reflection. To code the pulse, different shaped pulses are transmitted, using different time/voltage profiles to encode each separate transmitter. It is unlikely that this method is robust against superposition of two separate pulses, but information on simultaneous reception is not presented in the article. Once pulses have been identified, using template-matching, bearing information is computed using triangulation. All sonar sensors are still used in a strictly TOF manner, and no other data except for the pulse shape is used. Triangulation timing in these

DSPs is possible because the size of the sonar ring is fairly large (31cm diameter). Because of the wide baseline, timing precision requirements are not stringent. And yet, the entire system uses at least 1.25W of power just for computation of received pulses, given the components listed above. This neglects other necessary power components such as the analog front end for the receiving sensors. With this amount of power, a traditional phased array system based on digitization and DSPs could've been implemented without the need for pulse coding. The system is clearly elaborate and advanced, but arguably not very efficient in its processing.

For smaller applications with smaller baselines, the timing requirements become even more precise. Smaller robots with physical measurements much smaller than the 31cm Fazli ring would require at least an order of magnitude higher processing speed to accomplish triangulation the same way. At the same time, a smaller robot has less room for energy storage. These mutually exclusive realities strongly argue for alternate processing methods if sonar is to be effectively used on more miniature platforms.

Alternate methodologies attempting to mimic sonar systems in animals have been constructed to meet demands for efficiency. Shi and Horiuchi [75, 76] have produced binaural sonar receiving systems based on the processing methods used in the brain and nervous system of bats to interpret sonar returns. To even come close to the accuracy and energy-efficiency of a real bat sonar system would be a huge advance over any current man-made sonar system. Results of the system

show waveforms from its neural network processing elements which demonstrate the ability to discriminate between different reception bearing angles. As of yet the results are not quantitative, however. In the future, when this approach is used in a complete system it may be compelling. Currently, however, there are no published characteristics demonstrating system performance metrics, such as bearing precision, speed, and other figures of merit.

7.1.2 Using the STFAP

Starting from the bottom-up with an sensor system where the sensors are built to cooperate with each other is clearly the solution. And the STFAP does this. Instead of using large sensors to narrow the field of view, array processing is used with small wide angle sensors. Better than merely looking precisely in one narrow angle of view, array processing allows both a wide angle of view *and* discrimination between the bearing angles of returns within that view. Instead of reading one distance from envelope detection at each sensor, the full sensor waveforms are read for STFAP processing. This enables multiple ranges to be measured, and allows synchronous processing of the waveforms based on zero-crossings of the carrier frequency.

1° accuracy is demonstrated with extremely low power consumption (1mW). This is orders of magnitude smaller compared to the state of the art in the literature. With the linear array outlined previously, the current STFAP a compact 2-D

sensor that is extremely energy efficient. Combining two perpendicular linear arrays would allow full 3-D sensing of the environment, remaining extremely low power by present-day standards. For mobile robots, the demonstrated compactness, low-power usage, and accuracy of the STFAP allow the full promise of sonar sensing to be realized.

7.2 Security Sentry

For security applications, just as optical imaging with cameras is of fundamental importance, adding depth information from a 3-D sensor such as the STFAP can add vital information to applications designed to detect intruders.

One application of sonar where optical methods are clearly useless involves a device described by Wild et al. [77] for detecting people behind walls. By detecting AC signals through ultrasound returns, moving people and respiration of stationary people can be detected from faint and cluttered signals. The device described is designed to enable a team of Law Enforcement or Military personnel to see through a wall before entering a room. It allows vital information on the presence and location of friendly and/or hostile persons on the other side of the wall. At the same time, the ultrasonic frequencies used are not detectable to those being monitored without specialized equipment. The virtues of such a device to security personnel include: portability, ability to localize locations of people, and

fast update rate to accurately record motion such as respiration or heartbeats in order to detect people. STFAP circuitry is low-power, and consequently could remain very portable by necessitating a smaller battery for operation. At the same time, the fast operation and the bearing angle estimation allow fast determination of angle bearing or received pulses. It is thus very compatible with field use by Law Enforcement or Military personnel for this application.

Another application related to distributed sensing is a security outpost designed to detect intruders. Surveillance is often needed in isolated regions, such as border security around mobile military camps, or remote locations on national borders. Intelligence on movement of people is also needed in or around hazardous areas, such as disputed territory subject to warfare. There are numerous locations where it is advantageous to be able to track the movements of potentially hostile people without constant maintenance of the tracking method. In some extremely critical situations, no maintenance is possible after installation. In such locations, the lifetime of the device itself benefits from low-power operation. Wireless transmissions would allow monitoring from a distance the movements of people within the sonar array's field of view. The low-power operation of the STFAP is perfect for such a situation. Even less energy is necessary if the array is teamed with a "wake-up detector" detecting the presence of people. The sonar array could turn itself on only when 3-D precision is necessary, and efficiently track people within a given area. When the cost of maintenance or replacement is exposure to possibly

life-and-death situations, extended lifetime of a surveillance device is vital.

7.3 Biomedical / Retinal surgery

7.3.1 Background and Current Problems

Ultrasonic arrays have been used for medical imaging for decades. At the present time, however, most arrays used commercially are linear (1-D) arrays. Beam azimuth (along the line of the array) has been successfully electronically-steered in these devices, while altitude of the beam has been controlled by hand or mechanical means. Some of these linear arrays have additional capabilities for fixed beam-focusing in the altitude direction, but this usually does not include the ability to flexibly steer the beam. Such modified linear arrays are termed 1.5-D arrays.

Certain groups are in pursuit of true 2-D array technology. For medical applications, Smith and others in the Biomedical Engineering Department at Duke University have developed different 2-D array transducers using piezoelectric materials [78, 79], which is now in commercial use. Both kinds of sensor arrays are very promising because they allow full electronic beam steering in both directions.

Despite the promise, full 2-D arrays have not dominated medical ultrasonic imaging because of practical problems. The basis for most of the problems are interconnect issues. With frequencies of medical ultrasound in the megahertz range,

array elements need to be close together to adequately spatially sample space, with sensor pitch on the order of several hundred microns. With such tight spacing in two dimensions, connecting every element to cabling becomes difficult. The method of physically connecting each cable to each array element is challenging, as is keeping each line from each sensor electrically separate. Crosstalk between lines is a major design issue. In addition, connecting all of these multiple signal lines from the sensor location out of the body and back to the processing unit presents a long path, increasing the signal attenuation of each sensor line.

There can be considerable advantages to closely coupling the ultrasonic array processing with the sensing elements. Most importantly, the drawbacks discussed above result from interconnect issues and long cable lengths from the sensor to the processing. By integrating the processing with the sensors, issues relating to crosstalk and attenuation can be largely eliminated. Using Micro-Electro-Mechanical Systems (MEMS) sensors, there is the possibility of building ultrasonic transducers on the same silicon substrate as the processing electronics, or on compatible silicon substrates that can be bonded together with sub-mm-scale precision using "flip-chip" bonding techniques. Much work has been recently accomplished in this area, notably by Khuri-Yakob and associates [80]. In addition to increasing signal quality, this type of interconnect allows much more practical and repeatable interconnection, a practical advance in the construction of the unit.

Besides these primary benefits of closely-coupled sensing and processing, there

are system benefits as well. When the processing is at the site of the sensing, applications utilizing sonar in a feedback loop become possible. Since the physical length of the connections vastly decreases, the speed of the loop can vastly increase. Applications using direct feedback are currently being explored as active research projects in the Engineering Research Center for Computer Integrated Surgical Systems and Technology (CISST). In fact, the sonar array processor has been partially funded by the CISST ERC. The main thrust for research on miniature ultrasonic arrays has been microsurgery on the retina. As discussed previously in this thesis, retinal surgery demands near micron-scale precision of positioning and movement. One major problems for surgeons is hand tremor affecting precision of operating instruments. Another issue is that interaction with retinal tissue in most procedures remains below the threshold of touch, and visual feedback must be relied on completely to monitor and control surgical actions. A sonar system at the tip of the operating tool in the retina would have the ability to provide hard data about distance of the tool from the retinal tissue, rather than the inferred depth information received from surgical microscopes, and visible deformation of the tissue itself. In addition, by tracking landmarks in the retina, a direct measure of tool-tip motion due to surgeon tremor can be measured. The goal of future research in the CISST ERC is to use this 3-D information of tool position to compensate for tremor and protect against damage to the retina from accidental contact. Just as tracking movements using visual systems may allow for equal and opposite com-

pensating motions to steady a tool, so can surgeon tremor be eliminated with a sonar system. In addition to movement within the 2-D plane, with sonar depth information is also received, adding more information and the ability to steady a tool in another axis.

7.3.2 Using the STFAP

For processing to occur at the sensor, electronics will necessarily have to be very close to the body, or in many of the cases described previously, actually inside the body. Locations such as these present new requirements on the sensor processing solutions employed.

First of all, physical compactness is an issue. To be size-compatible with a micro-sonar array, the implementation of the processing electronics cannot be on the scale of centimeters, as many contemporary microprocessors or DSPs are today. The size of the STFAP chip is currently on the scale of millimeters, and is not expected to scale significantly larger with future generations. Small size is especially important for locations within the human body. For the retina this is an even stricter requirement. The total volume of the eye is quite small, as is the normal opening created for surgical tools to enter the eye.

Another issue for processing inside the body is power consumption. Virtually all power consumed by electronic circuits is given off as heat, and to be non-damaging to biological tissues this heat cannot be too extreme. Even if the size of

a traditional DSP could be shrunk down to a size small enough to be placed inside the body, if it still consumes over watts of power it will give that energy off as heat in a very small volume. This can be dangerous to tissues, and it may adversely affect operating conditions by heating up the immediate environment of the tool. The STFAP with its 1mW of power consumption is much more biologically-compatible.

Power consumption also relates to the size of the array. Using traditional phased-array techniques based on time-sampling the data, the close spacing of most biological sonar arrays and operational sonar frequencies of megahertz would require sampling rates nearing 100MHz. Any traditional processor dealing with frequencies this high would need a correspondingly higher power consumption compared with the relatively low frequencies used in robot navigation. The DSP-based processing of the Fazli ring described in Section 7.1.1 performed at 40kHz with 1.5W of power could easily scale to over 100W if the same processing were used. Obviously this is not practical. The processing in the STFAP, because it uses continuous-time processing and doesn't rely on time-sampled systems, will be able to scale to higher frequencies without a commensurate increase in power consumption. It should remain more biologically-compatible even at higher frequencies.

Low-power processing and biologically compatible implementation could enable even more visionary applications for sonar devices. For instance, implantable devices to measure blood flow or monitor an organ can be imagined. Implantable devices must be even more frugal with power than devices connected to the out-

side world by wires. They must use only the power from a battery, which may or may not be recharged using near-field magnetic power transmission through the skin. The power able to be transmitted through the skin is minor without causing damage: $10\mathrm{mW/cm^2}$. In addition to this, the bandwidth transmitted back through the skin to the outside world is also typically modest, less than 1Mbit. Raw data from multiple sonar transducers would present too much bandwidth, and so must be processed internally. This combination of necessary local processing and very low power usage presents very challenging design requirements, but future generations of the STFAP promise to make such devices possible.

Chapter 8

Conclusion

8.1 Efficient High-Performance Sensor Processing

The bulk of this thesis has been devoted to the detailing the performance of two types of sensor processing. The common thread has been sensing modalities designed from the ground up for efficiency, enabled by parallel access to the raw sensor data, and by taking advantage of mixed-signal application-specific processing. By translating the raw data directly to applicable forms for the computation involved instead of a completely generic digital representation, efficiency has been improved. As with any application-specific processing, the engineering trade-off has been made in favor of efficiency and performance, at the expense of some flexibility. The power- and size-efficient designs, greatly improve the sensing systems' appropriateness for certain applications, and enable their use in numerous appli-

cations not otherwise possible, as has been seen.

8.1.1 Autonomous Robotics

Autonomous robotics is one area which will greatly benefit from an improved vision system of DPCT vision sensor, and the special ability of the sonar processor to bring new clarity to the economical yet previously ambiguous sonar sensing modality. The sensor systems designed produce data that is far more valuable than a raw signal. Instead of raw analog sensor data, or high-bandwidth digital streams of raw sensor data, what these sensors provide can already be useful to a computer as features.

For a mobile robot, energy storage is already bulky and heavy. Clearly, efficient computation, allowing a robot to understand more of its environment using less power resources, is a valuable ability. Current robots are the size of an office trashcan, which allows for a certain amount of indoor movement. These "trashcan form-factor" robots are convenient for robot designers because they are able to accommodate large batteries and standard PC computers. For the next-generation of untethered smaller robots to become a reality, power efficiency will no longer be beneficial, but necessary.

Physical volume is also a valuable commodity for mobile robots. The single-chip implementation of my sensor processing greatly benefits the cramped spaces of robots. Instead of multiple boards devoted to DSPs, ADC chips, and associated

wiring, simplicity is provided by computational sensors. Again, what is a luxury for today's variety of mobile robots will become a fundamental requirement as sizes shrink.

Efficiency is useful to extend the useful range of present-day autonomous robots. To create robots the the size of a toaster or smaller, processing efficiency will need to be increase even further. My sensor systems provide this needed efficiency boost.

8.1.2 Biomedical Applications

For computer-assisted surgery, the role of the DPCT imager and sonar array processor become defined by their role in feedback loops. To compensate for unwanted motion and stabilize surgical tools, a measure of the unwanted motion is first needed. Measuring optically or by sonar tremor from the surgeon's hand to the surgical instrument is only part of the process. Completing the feedback loop successfully by causing actuators to create opposite motion and nullify the unwanted movement requires speed in the loop. The throughput of the traditional combination of sensor, ADC, and computer is not enough in this situation. The latency, or delay from the real-life event until the system has recognized the event can be more important than how many times per second this process is carried out. The normal method of serializing sensor data from parallel sources such as an imager or sonar array, digitizing the serial stream, and computing in sequential

steps, allows for much delay. Pipelining this process improves throughput but not total latency.

My sensor processing systems, because of their intimate access to the raw data, and parallel modes of computation, are able to notice real-life events and respond with very little latency. In addition, their throughput meets or exceeds the performance of their traditional counterparts.

8.1.3 Distributed Sensing

Distributed sensing nodes are independent sites for measuring the world that are likely to be in isolated areas. The main assumptions of my definition of a distributed sensor is that they need to run from a battery, need to communicate wirelessly, and have limited or no opportunity for maintenance. A typical example would be a space-based extra-terrestrial sensor, in orbit or on another planet.

For these applications, efficiency of size and power is of course key. Rising to equal importance with these abilities, is efficiency or compactness of the data resulting from the sensors. Wireless transmission is costly in the general sense of the word. Transmitting data wirelessly uses an amount of power directly related to the data rate. For limited power applications it makes no sense to transmit raw sensor data from the remote sensor to a central control unit. Thus some local processing is necessary to cull the useful features from the entire amount of data recorded from the sensors. When the sensing is combined with feature extraction, as with

my sensors, this process is intrinsic. From start to finish, the sensor and processing embody the efficiency philosophy, instead of working backward to attempt to make inefficient general-purpose methods less wasteful.

8.2 Sensor Fusion / Future Work

As have seen, individually these sensors are an effective way to allow different views of the world in a way that is compatible with resource-limited applications. The greater promise lies in using them in a coordinated way to extract even richer 3-D information about the world. Using protocols and methodologies already developed, both sensors are ready to be interfaced to a central processor so their data can be fused.

The most obvious advantage to using multiple sensors is that deficiencies in one sensing modality are often not deficiencies in the other. The "blind spots" of one are likely not a problem for the other.

8.2.1 Complementary Strengths and Weaknesses

8.2.1.1 Depth Information

The DPCT visual sensor, for instance, can possibly perceive depth information. Various schemes exist for one camera depth. Moving the camera and keeping track of relative motions of objects within view can be used to receive cues for

depth. Closer objects appear to move faster across the focal plane than far objects. A lens aperture set to achieve short depth-of-field can be used with active focusing to deduce the depth of objects, by noting which parts of the image are currently in focus at a particular focus setting. Finally, two cameras can sense depth using stereo vision techniques that have been extensively developed and detailed over the years.

These methods work to a better or worse degree. Some information about depth can surely be gleaned by any of these approaches, but the DPCT imager is definitely not optimized for these tasks. Tracking a full-motion vector field for the image to deduce relative depth is beyond the scope of the object tracking the DPCT imager was designed for. Similarly, measuring contrast information to deduce the depth of objects from lens focus implies another full subsystem in operation. It's possible that a computer could process the APS images to provide these extra functionalities. Indeed, one feature of having true image information output from my imager in addition to computed features is the extensibility of the architecture—a system designer has free access to high-fidelity image data to process, giving the ability to create more advanced features when needed. For the DPCT imager, however, where emphasis is placed on efficiency, this should be viewed only as a second choice if no other methods are available.

Depth from stereo vision represents a special case. Stereo view of a generic scene with the DPCT imager would pose a classical computer vision task. Match-

ing features in one camera to the same features in the other remains a computationally-intensive task. From two cameras' APS data, all of the attendant pitfalls and compute cycles for any other types of cameras are no different. The outlook for stereo vision greatly improves for dual DPCT imagers, however, with the introduction of active beacons (blinking LEDs) into the scene. Because this imager is optimized for the tracking of the position of temporal changes, a blinking LEDs position can be computed with all of the speed, low latency, and efficiency inherent to the DPCT imager's optimized design. This kind of stereo vision would allow pinpoint locations of active beacons to be read out much faster than a full frame-rate (30fps) commercial camera, at the 180pos./s or more of the position tracking subsystem. In addition, no extra latencies would be present from LED stimulus to position output from each camera. The resulting calculations of the epipolar stereo geometries represent a much smaller task than feature matching between images.

However, artificial beacons in a scene remain a special case. There may be many situations where one is not so fortunate as to be able to place landmarks ahead of time on features of interest. For these situations, all that is available are the aforementioned alternative visual methods for depth perception, which are clearly difficult and non-optimal. Sonar, however, is a sensing modality that *is* optimized for depth perception. With sonar, it is possible to get a much denser depth map than with many visual methods. Beacons represent only a few points in the scene, and other feature-matching methods are able to capture limited numbers of corre-

spondence points depending on the amount of processing involved. Because the STFAP sonar has bearing angle directionality in addition to depth, the possibility exists with extra processing to register a sonar depth profile with a visual image. The case for a sonar/visual combination improves when using two 1-D arrays or a 2-D array, so that both altitude and azimuth polar angles are available, allowing a full 3 axes to match with the 2-D representation from the imager.

8.2.1.2 Active Versus Passive Sensing

Another characteristic distinguishing one sensing method from the other is the method of illumination. In the case of the visual imager, it is assumed that the image will be formed from ambient light already present in the scene. For sonar, a basic requirement is to provide the illumination, in this case a burst of ultrasonic energy. Each presents advantages and disadvantages when compared to the other.

For stealth or secrecy of sensing, it is obvious that passive sensing is preferable to active sensing. A device should ideally not create any energy that can be noticed, it should merely record the energy already present in its environment. For stealthy applications, a visual imager may have the advantage because it is passive. However, it all depends on whether people who could notice an active sensor are looking for the emissions it gives off. Ultrasonic sounds, being above human hearing, are not able to be sensed without special equipment. Thus, for unsuspecting people not specifically looking for ultrasonic frequencies in their en-

vironment, sonar could be considered somewhat stealthy. But the fact remains that creating no output is a sure way to remain unnoticed. If the extra depth precision of sonar is needed in a secret application, one could imagine predominantly using the visual system, and adding the ultrasonic system for short periods of time only when absolutely necessary. This would restrict the amount of time ultrasonic bursts are being created, helping to preserve secrecy.

Another contrast of passive vs. active sensing techniques is the amount of energy necessary to provide the illumination. A sonar burst output typically uses far more energy than reception and processing of the returns. In a Polaroid 6500 stock sonar ranging module, for instance, transmission requires a factor of 20 more energy than reception [81]. To produce sonar outputs all the time thus presents a non-negligible waste of energy in the presence of nothing interesting to observe. Again, a hybrid system using a visual sensor set to wait for something "interesting" to happen before activating the active sonar system could solve this problem.

8.2.1.3 Specularity

The vastly different frequencies/wavelengths of the observables of each sensor system have a significant effect on their ability to view the world. Surfaces can reflect waves in basically two ways. They can act more like a perfect mirror, where the incident angle to normal is the same as the exit angle to normal, like the motion of a billiard ball off of a bumper. Or they can act in a more diffusing manner, re-

flecting incoming waves with varying energies in all directions. If a surface is close to a perfect mirror, it is termed specular, and if it tends to reflect in a diffuse way, it is termed Lambertian. Generally speaking, the physical size of the irregularities on a surface compared to the wavelength of the incoming wave will predict whether the surface appears specular or Lambertian at that wavelength. Rough surfaces with random irregularities larger than the incoming wavelength will tend to scatter incoming waves. Smooth surfaces with smaller irregularities will tend to reflect more rays in a mirror-like way.

Because the wavelength used for sonar in air is approximately 4 orders of magnitude larger than the wavelength for visible light, many more surfaces interact with sonar in a specular way than visible light. The wavelength of a 40kHz sound wave in air is about 8.5mm, and the wavelength of red visible light is about 700nm. A surface with random features on the order of millimeters is very rough indeed. Most surfaces found in a typical man-made environment are much smoother than this. Hence almost everything found indoors, and even some natural objects in outdoor environments will appear specular to sonar in air. In Biomedical applications where the sonar frequencies used are in between 1MHz and 10MHz, the sonar wavelengths are between $34\mu\mathrm{m}$ and $340\mu\mathrm{m}$, much shorter and thus fewer surfaces appear specular.

The result of this is that most surfaces viewed with sonar in air may not reflect very much sound energy back at the sonar device, but will instead reflect the sound

on away from the transmitter/receiver unit. At the same time, very few surfaces are specular to light, because most do not have random surface features smaller than 700nm. Thus light falling on most surfaces will be reflected in all directions, allowing a camera in almost any location to see objects under visible light. One of the few specular surfaces is an ordinary mirror—one sees objects reflected in the mirror, but it is very difficult to examine the mirror material itself.

Specularity is one of the common challenges with using sonar effectively. The STFAP solves the problem of discriminating between single reflections and multiple-bounce reflections. But a more difficult problem is how to accomplish sensing when very little of the incident sonar pulse comes back to the receiving unit at all. The solution to this problem is to also use visible light imaging in tandem, as a sanity check to confirm sonar reports of the absence of objects. For moving objects which are usually of interest because they represent the dynamic portion of a scene, the DPCT imager is especially well-suited.

8.2.2 Data Fusion Methods

Using two sensors together in a really coordinated way is not trivial. In fact it is a whole engineering field called Sensor Fusion. Obviously, the exact method for combining the data from sonar and visible imaging sensors is not obvious and qualifies for an entire study in itself. At the same time, some basic strategies for blending the data from these particular sensors can be described. Because my sen-

sors have some non-traditional architectures, their unique attributes can be used to advantage.

The first characteristic of note for both sensor processing schemes is that there is a definite progression of the representation of data from most raw and parallel, to less parallel, to a final one or few features output as measurement. Normally, the goal is to keep distilling the representation into the richest, least entropic representation possible to be as friendly as possible to the next subsystem that will act as a consumer of data. However, this may not always be in the best interest of a processor unit attempting to carry out sophisticated data fusion. It may want access to a slightly lower-level version of the dataset coming from each sensor. An advantage of the sensor processors described in this thesis is that there are a few possible levels of data representation available. Raw data from all constituent sensor parts is not immediately serialized to forward to a microprocessor. Along the way to the final answer, my processing carries out data processing of parallel data in a way that retains some of the parallel structure which may be useful for a later data fusion processor.

Internally, my sonar processor chip uses 16 separate spatiotemporal filter blocks that can be combined to one or four output values. Each of these filters represents added value of processing when compared to the raw microphone data. However, a view of these filters in parallel allows a segmented view of the spatiotemporal spectrum that is not available from reading the final bearing angle estimate. The

responses from these spatiotemporal filters can thus provide valuable insight for an external processor able to make use of them.

Similarly, my temporal-change pixels in the DPCT imager individually signal change in incident light, row and column sensors flag each row or column for one or more pixels changed in their group, and finally the output centroid voltages pool rows and columns together for simple x and y values. An external processor desiring access to slightly less-processed data could be allowed to access the raw column and row flags. With additional modification of the imager array, full access to every pixel event could be provided.

Both sensors include some asynchronous aspects of their operation and data production that could be used effectively with an asynchronous communication system. From an high-level perspective there are some very regular events which initiate processing in both sonar and imager systems. The APS array is fully synchronous and scanned, and it doesn't provide computation. The centroid-tracking subsystem of the imager starts with an reset applied at regular intervals. And for the sonar system, the original sonar output ping is the beginning of all possible computation. While the initiating events of these processing systems are regular, repeated events, there is nothing that requires the consumer of their data to be in lockstep synchrony with these events. In fact, the wait for data output can vary widely for both sonar processor and centroid imager, or be absent altogether if there are no observables to report. The wait to see if data exists can be millisec-

onds, to see at most dozens of bytes of data. Such sparseness of data could be capitalized upon by an asynchronous method of transmission.

Parallel sensor processing units producing parallel data suggests that a parallel implementation of a data fusion processor would also be advantageous. The structure of the data itself has spatial relationships that may be successfully exploited. For the imager, the spatial relationships are a direct mapping from the focal plane. Adjacent rows and columns, or more specifically regions of pixels are related to each other and need to be processed in a way that is sensitive to locality. Also, segments of the sonar processor's spatiotemporal space need to be analyzed by the responses of locations in the 2-D space. Instead of making these spatial relationships a virtual feature of the computation, real parallel processing blocks could speed up computation, working simultaneously by taking advantage of the repeatable spatial nature of the input data. In a digital processor paradigm, parallel processors could be used. And because the computation will likely be very similar, optimization to a Single Instruction Multiple Data (SIMD) methodology may be possible, where the same exact code runs at the same time on the different segments of the dataset. If the output of the fusion processor needs to interface to a more biologically-inspired system, a neural network could be the chosen spatially-savvy method of computation.

8.3 Summary

I have detailed the design, construction, and successful testing of two different sensing systems. The systems are application-specific, optimized to carry out useful sensor computations while using the least amount of power and external circuits. The first is a complete imager with facilities for high-quality image reproduction of the scene at normal video frame rates (30fps) and even faster tracking of the position of moving objects in the scene. The second system is the processing system for a miniature sonar array using a novel processing scheme.

The imager has demonstrated the normal output of images based on integrating pixels, allowing for improved immunity to noise over pixels based on instantaneous photocurrent. This classic imaging functionality with noise-reducing Double Sampling circuitry to subtract reset noise has been demonstrated at a frame rate fast enough for full-motion video reproduction. In addition to the image reproduction ability of the imager, the focal plane contains additional pixels optimized for sensing temporal change in incident light. After sampling their light level during frame reset, they are able to signal a change from this reference level at any time. Temporal changes from pixels are able to be communicated to circuitry on the edges of the array as they happen, and latches along the rows and columns keep track of these events as they happen. While these changes are occurring, instantaneous computation of the centroid of the rows and columns showing changes is occurring. Because computation of all of these parts occurs without

scanning and events are flagged as soon as the incident light changes, computation of movement position can happen very quickly, and with virtually no latency. In fact, centroid tracking throughput has been demonstrated at 180pos./s, with less latency than the time for one position to be reported, or 1/180seconds. The high update speed and near-zero latency uniquely qualify this imager to be used in an optical feedback loop. All of this functionality is carried out with extremely low power-consumption of 2.6mW, allowing the imaging and tracking functions to be used on platforms which have limited power resources.

The sonar array processor chip successfully adds bearing angle to the traditional range computation of sonar. It uses a novel method of processing a microphone array, neither like delay-and-sum nor fully digital DSP-based architectures. The processing is based on spatiotemporal frequency filtering. By applying spatial kernels instantaneously to microphone signals and time-filtering the spatial filter outputs, a bearing estimate can be quickly calculated from only a few zero-crossings of an ultrasonic carrier in a sonar pulse. The processing is already compatible with a small microphone array (26.5mm total baseline), which would likely be inaccessible to simple triangulation by from traditional time-of-flight sensors. Part of the key to this success is the synchronous processing based on the ultrasonic carrier wave, not the ill-defined pulse envelope. The resulting bearing estimate is enabled to be far more accurate than calculation using combinations of TOF sonar sensors. Additionally, since processing uses continuous-time and not

sampled-time methods, the baseline of the array can be made smaller without an increase in a global clock. In time-sampled methods, the clock must be run ever faster to prepare for the case of most precision, which can scale to hundreds of megahertz or even gigahertz as the array is scaled smaller. For my continuous-time approach, there is no need for a constantly running global clock, so power isn't wasted on high-power circuits built to be stable for the worst case short clock period, and power isn't used when no signals are transitioning. Instead, circuit precision is the limiting factor on system precision. This trade-off allows ever-smaller arrays to be approached without a microwave-frequency clock, using only as much power as is necessary.

In addition, these two sensing systems output data streams are amenable to combining using data fusion techniques. Their frugal needs, compact size, and consideration for downstream processing enable them to be an enabling technology for biomedical applications, distributed sensing, and truly autonomous small robots.

Bibliography

[1] U. Mallik, M. Clapp, E. Choi, G. Cauwenberghs, and R. Etienne-Cummings, "Temporal change threshold detection imager," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, vol. 48, 2005, pp. 362–363.

[2] P. I. Corke, P. A. Dunn, and J. E. Banks, "Frame-rate stereopsis using non-parametric transforms and programmable logic," in *Proc. IEEE International Conf. on Robotics and Automation*, vol. 3, May 1999, pp. 1928–1933.

[3] J. Woodfill and B. Von Herzen, "Real-time stereo vision on the PARTS reconfigurable computer," in *Proc. of the IEEE Symposium on FPGAs for Custom Computing Machines*, Apr. 1997, pp. 201–210.

[4] L. Warren, "Hull-mounted sonar/ship design evolution and transition to low-frequency applications," *IEEE J. Oceanic Eng.*, vol. 13, no. 4, pp. 296–298, Oct. 1988.

[5] P. N. T. Wells, "Current status and future technical advances of ultrasonic imaging," *IEEE Eng. Med. Biol. Mag.*, vol. 19, no. 5, pp. 14–20, Sep/Oct 2000.

[6] R. Kuc and B. Barshan, "Navigating vehicles through an unstructured environment with sonar," in *Proc. of the IEEE International Conf. on Robotics and Automation*, vol. 3, May 1989, pp. 1422–1426.

[7] H. Griffiths, J. Griffiths, C. Cowan, R. Eiges, and T. Rafik, "Processing techniques for circular sonar arrays," in *Sixth International Conf. on Electronic Engineering in Oceanography*, July 1994, pp. 67–72.

[8] K. Nagatani, H. Choset, and N. Lazar, "The arc-transversal median algorithm: an approach to increasing ultrasonic sensor accuracy," *IEEE International Conf. on Robotics and Automation*, vol. 1, pp. 644–651, 1999.

[9] J. Leonard and H. Durrant-Whyte, "Application of multi-target tracking to sonar-based mobile robot navigation," in *Proc. of the 29th IEEE Conf. on Decision and Control*, vol. 6, Dec. 1990, pp. 3118–3123.

[10] D. Bank, "A novel ultrasonic sensing system for autonomous mobile systems," vol. 2, no. 6, pp. 597–606, Dec. 2002.

[11] K. E. Thomenius, "Evolution of ultrasound beamformers," in *Proc. IEEE Ultrasonics Symposium*, vol. 2, Nov. 1996, pp. 1615–1622.

[12] Y. Mo, T. Tanaka, K. Inoue, K. Yamashita, and Y. Suzuki, "Front-end processor using BBD distributed delay-sum architecture for micromachined ultrasonic sensor array," *J. Microelectromech. Syst.*, vol. 12, no. 4, pp. 506–512, Aug. 2003.

[13] G. Hampson and A. Paplinski, "Hardware implementation of an ultrasonic beamformer," in *Proc. of IEEE Speech and Image Tech. for Computing and Telecom. (TENCON '97)*, vol. 1, Dec. 1997, pp. 227–230.

[14] A. Paplinski, N. Bhattacharjee, and C. Greif, "Rotating ultrasonic signal vectors with a word-parallel CORDIC processor," *Proc. Euromicro Symp. on Digital Systems, 2001.*, pp. 254–261, 2001.

[15] M. A. Clapp and R. Etienne-Cummings, "A dual pixel-type array for imaging and motion centroid localization," *IEEE Sensors J.*, vol. 2, no. 6, pp. 529–548, Dec. 2002.

[16] ——, "A dual pixel-type imager for imaging and motion centroid localization," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 3, May 2001, pp. 501–504.

[17] U. Mallik, M. Clapp, E. Choi, G. Cauwenberghs, and R. Etienne-Cummings, "Temporal change threshold detection imager," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, vol. 48, 2005, pp. 362–363.

[18] M. Clapp and R. Etienne-Cummings, "Tracking multiple targets in 3d," in *Proc. of the 35th Conf. on Information Science and Systems*. Baltimore, MD: Johns Hopkins University, Mar. 2001, invited.

[19] R. Etienne-Cummings, V. Gruev, and M. A. Clapp, "High density focal-plane image processing," in *Proc. SPIE*, 2001, invited Paper.

[20] ——, "High performance biomorphic image processing under tight space and power constraints," *Autonomous Robots*, vol. 11, no. 3, pp. 227–232, Nov. 2001.

[21] M. A. Clapp and R. Etienne-Cummings, "Single ping-multiple measurements: sonar bearing angle estimation using spatiotemporal frequency filters," *IEEE Trans. Circuits Syst.*, 2005, (submitted).

[22] ——, "Sensing signal input bearing to sensor array using velocity-sensitive spatiotemporal filters," *Electronics Letters*, vol. 40, no. 3, pp. 211–212, Feb. 2004.

[23] ——, "Bearing angle estimation for sonar micro-array using analog vlsi spatiotemporal processing," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 4, May 2004, pp. 884–887.

[24] ——, "Ultrasonic bearing estimation using a MEMS microphone array and spatiotemporal filters," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 1, May 2002, pp. I–661–I–664.

[25] R. Etienne-Cummings and M. A. Clapp, "Architecture for source localization with a linear ultrasonic array," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 1, May 2001, pp. 181–184.

[26] S. Mendis, S. E. Kemeny, and E. R. Fossum, "CMOS active pixel image sensor," *IEEE Trans. Electron Devices*, vol. 41, pp. 452–453, Mar. 1994.

[27] S. K. Mendis, S. E. Kemeny, R. C. Gee, B. Pain, C. O. Staller, Q. Kim, and E. R. Fossum, "CMOS active pixel image sensors for highly integrated imaging systems," *IEEE J. Solid-State Circuits*, vol. 32, pp. 187–197, Feb. 1997.

[28] C. Koch and H. Li, Eds., *Vision Chips: Implementing Vision Algorithms with Analog VLSI Circuits*. Los Alamitos, CA: IEEE Computing Society Press, 1995.

[29] S. DeWeerth, "Analog VLSI circuits for stimulus localization and centroid computation," *International Journal of Computer Vision*, vol. 8, no. 2, pp. 191–202, 1992.

[30] Z. Kalayjian, "VLSeye: Optoelectronics vision and image processing," Ph.D. dissertation, Johns Hopkins University, Baltimore, MD, 1999.

[31] Y. Bar-Shalom, H. M. Shertukde, and K. R. Pattipati, "Use of measurements from an imaging sensor for precision target tracking," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 25, pp. 863–872, Nov. 1989.

[32] M. Barbaro, P. Burgi, A. Mortara, P. Nussbaum, and F. Heitger, "A 100×100 Pixel silicon retina for gradient extraction with steering filter capabilities and temporal output coding," *IEEE J. Solid-State Circuits*, vol. 37, pp. 160–172, Feb. 2002.

[33] G. Indiveri, "Neuromorphic analog VLSI sensor for visual tracking: Circuits and application examples," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 1337–1347, Nov. 1999.

[34] K. Cameron, J. Canaris, and S. Whitaker, "A video image processor for real-time centroid calculation," in *Proc. of the Fifth Annual IEEE International ASIC Conf. Exhibit*, 1992, pp. 201–204.

[35] D. L. Standley, "An object position and orientation IC with embedded imager," *IEEE J. Solid-State Circuits*, vol. 26, no. 12, pp. 1853–1859, Dec. 1991.

[36] M. H. White, D. R. Lampe, F. C. Blaha, and I. A. Mack, "Characterization of surface channel CCD image arrays at low light levels," *IEEE J. Solid-State Circuits*, vol. SC-9, pp. 1–13, Feb. 1974.

[37] C. C. Enz and G. C. Temes, "Circuit techniques for reducing the effects of op-amp imperfections: Autozeroing, correlated double sampling and chopper stabilization," *Proc. IEEE*, vol. 84, pp. 1584–1614, Nov. 1996.

[38] K. Boahen, "Retinomorphic chips that see quadruple images," in *Proc. of the Seventh International Conf. on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems*, Granada, Spain, Mar. 1999, pp. 12–20.

[39] K. A. Boahen, "Point-to-point connectivity between neuromorphic chips us-

ing address events," *IEEE Trans. Circuits Syst. II*, vol. 47, pp. 416–434, May 2000.

[40] E. Culurciello, R. Etienne-Cummings, and K. Boahen, "Arbitrated address event representation digital image sensor," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2001, pp. 92–93.

[41] H. Barlow, *The Senses: Physiology of the Retina*. Cambridge, MA: Cambridge Univ. Press, 1982.

[42] R. H. Nixon, S. E. Kemeny, B. Pain, C. O. Staller, and E. R. Fossum, "256×256 CMOS active pixel sensor camera-on-a-chip," *IEEE J. Solid-State Circuits*, vol. 31, pp. 2046–2050, Dec. 1996.

[43] H. Tian, B. Fowler, and A. El Gamal, "Analysis of temporal noise in CMOS photodiode active pixel sensor," *IEEE J. Solid-State Circuits*, vol. 36, pp. 92–101, Jan. 2001.

[44] P. E. Allen and D. R. Holberg, *CMOS Analog Circuit Design*. Oxford, U. K.: Oxford University Press, 1987.

[45] C. D. Motchenbacher and J. A. Connelly, *Low-Noise Electronic System Design*. New York: Wiley, 1993.

[46] R. Etienne-Cummings, V. Gruev, and M. Clapp, "High performance biomor-

phic image processing under tight space and power constraints," *Autonomous Robots*, vol. 11, no. 3, pp. 227–232, Dec. 2001.

[47] D. Tomlin, "Shrinking the limits: proprioceptive techniques for microsurgical skills augmentation," Master's thesis, Johns Hopkins University, Baltimore, MD, 2001.

[48] R. Taylor, P. Jensen, L. Whitcomb, A. B. R. Kumar, D. Stoianovici, P. Gupta, Z. Wang, E. deJuan, and L. Kavoussi, "A steady-hand robotic system for microsurgical augmentation," *The International Journal of Robotics Research*, vol. 18, no. 12, pp. 1201–1210, 1999.

[49] Micron Technology, Inc. (2005, July) MT9M011 data sheet. MT9M011.pdf. [Online]. Available: http://download.micron.com/pdf/datasheets/imaging/

[50] TransChip. (2005, July) Codercam TC5747 product brief. TC5747_Product_Brief_30March05.pdf. [Online]. Available: http://www.transchip.com/objects/

[51] S. Hamami, L. Fleshel, and O. Yadid-Pecht, "CMOS APS imager employing 3.3V 12-bit 6.3 MS/s pipelined ADC," in *Proc. of the International Symposium on Circuits and Systems*, May 2004, pp. 960–963.

[52] Z. Zhou, B. Pain, and E. Fossum, "A CMOS imager with on-chip variable reso-

lution for light-adaptive imaging," in *IEEE Intl. Solid-State Circuits Converence Digest of Technical Papers*, Feb. 1998, pp. 174–175.

[53] B. Pain, G. Yang, B. Olson, T. Shaw, M. Ortiz, J. Heynssens, C. Wrigley, and C. Ho, "A low-power digital camera-on-a-chip implemented in CMOS active pixel approach," in *Proc. of the Twelfth International Conf. On VLSI Design*, Jan. 1999, pp. 26–31.

[54] Texas Instruments, Inc. (2005, Oct.) Msp430x21x1 mixed signal microcontroller datasheet. msp430f2131.pdf. [Online]. Available: http://www-s.ti.com/sc/ds/

[55] G. Welch and E. Foxlin, "Motion tracking: no silver bullet, but a respectable arsenal," *IEEE Computer Graphics and Applications*, vol. 22, no. 6, pp. 24–38, Nov/Dec 2002.

[56] G. Welch, G. Bishop, L. Vicci, S. Brumback, and K. Keller, "High-performance wide-area optical tracking: The hiball tracking system," *Presence: Teleoperators and Virtual Environments*, vol. 10, no. 1, pp. 1–21, Feb. 2001.

[57] M. Clapp and R. Etienne-Cummings, "Tracking multiple targets in 3d," in *Proc. of the 35th Conf. on Information Science and Systems*. Baltimore, MD: Johns Hopkins University, Mar. 2001, invited.

[58] S. Kawahito, M. Yoshida, M. Sasaki, K. Umehara, D. Miyazaki, Y. Tadokoro,

K. Murata, S. Doushou, and A. Matsuzawa, "A CMOS image sensor with analog two-dimensional DCT-based compression circuits for one-chip cameras," *IEEE J. Solid-State Circuits*, vol. 32, no. 12, pp. 2030–2041, Dec. 1997.

[59] S. H. Hong and W. Yang, "An embeddable low power SIMD processor bank," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, 2000, pp. 192–193.

[60] W. D. Leon, S. Balkir, K. Sayood, and M. W. Hoffman, "Charge-based prediction circuits for focal plane image compression," in *Proc. of the International Symposium on Circuits and Systems*, vol. 4, May 2004, pp. 936–939.

[61] R. Etienne-Cummings, J. Van der Spiegel, and P. Mueller, "Hardware implementation of a visual-motion pixel using oriented spatiotemporal neural filters," *IEEE Trans. Circuits Syst. II*, vol. 46, no. 9, pp. 1121–1136, Sept. 1999.

[62] K. Aizawa, H. Ohno, T. Hamamoto, M. Hatori, and J. Yamazaki, "A novel image sensor for video compression," in *Proc. IEEE International Conf. on Image Processing*, vol. 3, no. 13–16, Nov. 1994, pp. 591–595.

[63] W. D. Leon, S. Balkir, K. Sayood, and M. W. Hoffman, "A cmos imager with pixel prediction for image compression," in *Proc. of the International Symposium on Circuits and Systems*, vol. 4, May 2003, pp. 776–779.

[64] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[65] Y.-J. Chiu and T. Berger, "A software-only videocodec using pixelwise conditional differential replenishment and perceptual enhancements," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 3, pp. 438–450, Apr. 1999.

[66] J. Maunsell and D. Van Essen, "Functional-properties of neurons in middle temporal visual area of the macaque monkey 1. selectivity for stimulus direction, speed, and orientation," *Journal of Neurophysiology*, vol. 49, no. 5, pp. 1127–1147, 1983.

[67] E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *Journal of the Optical Society of America A*, vol. 2, no. 2, pp. 284–299, Feb. 1985.

[68] A. B. Watson and A. J. Ahumada, Jr., "Window of visibility: A psychophysical theory of fidelity in time-sampled visual motion displays," *Journal of the Optical Society of America A*, vol. 3, no. 3, pp. 300–307, Mar. 1986.

[69] D. J. Heeger, "Model for the extraction of image flow," *Journal of the Optical Society of America A*, vol. 4, no. 8, pp. 1455–1471, Aug. 1987.

[70] L. T. Bruton and N. R. Bartley, "Three-dimensional image processing using the concept of network resonance," *IEEE Trans. Circuits Syst.*, vol. CAS-32, no. 7, pp. 664–672, July 1985.

[71] M. A. Clapp and R. Etienne-Cummings, "Ultrasonic bearing estimation using

a MEMS microphone array and spatiotemporal filters," in *IEEE International Symposium on Circuits and Systems*, vol. 1, 2002, pp. I–661–I–664.

[72] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Trans. Robot. Automat.*, vol. 3, no. 3, pp. 249–265, June 1987.

[73] H. Choset, K. Nagatani, and N. A. Lazar, "The arc-transversal median algorithm: a geometric approach to increasing ultrasonic sensor azimuth accuracy," *IEEE Trans. Robot. Automat.*, vol. 19, no. 3, pp. 513–521, June 2003.

[74] S. Fazli and L. Kleeman, "A real time advanced sonar ring with simultaneous firing," in *Proc. IEEE/RSJ International Conf. on Intelligent Robots and Systems*, vol. 2, 2004, pp. 1872–1877.

[75] R. Z. Shi and T. K. Horiuchi, "A VLSI model of the bat lateral superior olive for azimuthal echolocation," in *Proc. of the International Symposium on Circuits and Systems*, vol. 4, May 2004, pp. 900–903.

[76] ——, "A VLSI model of the bat dorsal nucleus of the lateral lemniscus for azimuthal echolocation," in *Proc. of the International Symposium on Circuits and Systems*, May 2005, pp. 4217–4220.

[77] N. C. Wild, F. Felber, M. Treadaway, F. Doft, D. Breuner, and S. Lutjens, "Ultrasonic through-the-wall surveillance system," in *Enabling Technologies for Law Enforcement and Security*, 2001.

[78] S. W. Smith, H. G. P. Jr., and O. T. von Ramm, "High-speed ultrasound volumetric imaging system. i. transducer design and beam steering," *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, vol. 38, no. 2, pp. 100–108, 1991.

[79] S. W. Smith, G. Trahey, and O. T. von Ramm, "Two dimensional arrays for medical ultrasound," *Ultrasonic Imaging*, vol. 14, pp. 213–233, 1992.

[80] O. Oralkan, A. S. Ergun, C.-H. Cheng, J. A. Johnson, M. Karaman, T. H. Lee, and B. T. Khuri-Yakub, "Volumetric ultrasound imaging using 2-D CMUT arrays," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 50, no. 11, pp. 1581–1594, Nov. 2003.

[81] SensComp, Inc. (2005, Aug.) 6500 data sheet. 6500%20module%20spec.pdf. [Online]. Available: http://www.senscomp.com/specs/

Vita

Matthew A. Clapp received his B.S. degree in Electrical Engineering from the California Institute of Technology in 1996. For two years he worked at Synaptics, Inc., chiefly designing mixed-signal ICs. He received his M.S.E. in electrical engineering from Johns Hopkins University in 2000. His research focuses on mixed-signal VLSI design, including intelligent sensors, which tightly couple sensing of the real world with processing. After leaving Johns Hopkins University, he will work for Agere Systems in San Jose, California as a Member of the Technical Staff.