



Simplified ASL Hand Gesture Recognition Machine Learning

01.11.2017

Joseph Ellsworth

CTO Bayes Analytic LLC

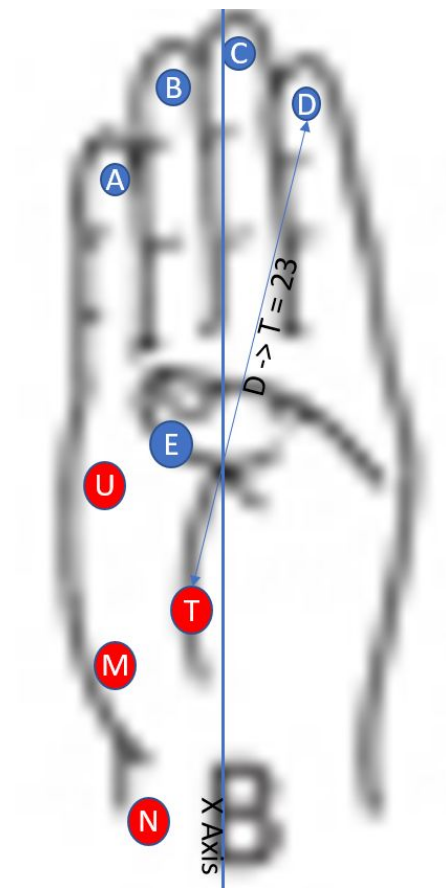
<http://BayesAnalytic.com>

Overview

This document is the result of research into a simplified way of thinking about hand shapes to describe and recognize static ASL hand gestures.

The goal was to develop a set of inputs for training and testing a ML engine that was dramatically simplified compared to the full body state from VR sensors which may include hundreds of position, rotation, momentum and acceleration sensors.

The assertion is that we would use a pipeline processor as described in our Architectural proposal for VR gesture recognition with one stage to reduce the input set to a simplified input set used by this algorithm.



I wanted to test was whether I could intuit the hands overall shape by measuring the fingers position relative to the palm. I suspect that to make this truly useful would need also know orientation of hand relative to body. Position of hand relative to body. This would be limited to recognition of gesture in a single frame.

The Distance from the tip of each finger is measure to a common point on on the hand as illustrated.

This would give a single frame measurement of:

A To T, B To T, C to T, D to T, E to T Measurement of M to N and T to N can be used to measure wrist movement which is needed for some gestures.

If a single point hand measurement is not adequate due to things such as a curved finger that is spread yielding the

same measurements then we can add a separate triangulation point to include

A to U, B to U, C to U, D to U and possibly to one or more axis such as A TO X, B to X, C to X, D to X, E to X.

There may be more optimal positions for T and U these were chosen as proximal positions to test the theory.

Essential Theory:

- A given gesture is composed shape of the hand which can be deduced from by determining how much each finger is bent.
- If a given finger such as A is curved over as in a fist it will be closer to point T than if it was extended.

- In addition if a the thumb is straight up the side of the hand then the distance from E to T will be greater.
- The same gesture from a small hand should be recognizable even if trained from a large hand. This means the distances measured should be relative from measurements if the hand was flat with all fingers together.

Calibration:

To make the system useful it should work across a wide variety of hand sizes. While some people may only form the gestures loosely.. This means we need a way to compute the position of the fingers relative to their maximum distance from the measurement point. This could be complicated if the only thing we have available was a closed fist. A simplifying assumption will be used that we are allowed to measure the flat fully extended position of the sensors. Or we can deduce this from the distance from T to U.

Rationalizing numbers:

To support variable sized hands the important value is the current distance relative the calibration distance. As such the numbers will be supplied to the engine in the form of $\text{Current}(A \text{ to } T) / \text{Calibration}(A \text{ to } T)$ providing a ratio metric number between 0 and 3 with most measurements being between 0 and 1.

Testing strategy

Develop manual measurements from various dictionary representation to feed in as training data then use measurements from a separate set

Convert Data to ML friendly representation

One of the secrets of many machine learning projects is coercing the existing data into a shape form that makes it easier to discover the signal in the noise. For example a pure stock price from bar to bar is information but an alternate way of looking at the same data is the percentage above the lowest price over the last 30 days.

Most ML engines are not capable of inventing these alternate ways of looking at the data on their own. In this instance I suspect that looking at each measurement as a fraction of the calibration measurement for the same sensor will yield more useful data than the straight measurements especially when the training data needs to be used to recognize gestures from people with many sizes of hands.

As with all suspicions it needs to be tested which I will do by feeding the engine training data and testing the recognition capability with both the raw data and the ratio data.

Finding unique ways to represent the existing data such that it yields the best results is one of the areas in ML that is still represent a lot of art and instincts on the part of ML practitioners.

Limitations

1. Some gestures are only meaningful when the hand is rotated toward or away from the body. It would require an additional sensor to detect that rotation.
2. Some gestures are meaningful only when made relative to the body trunk or head. We would need additional sensors to measure those.
3. This test is only intended to classify static hand gestures made with one or both hands and will not attempt to measure gestures that require tracking motion between frames. It may be possible to extend this work to include multi-frame gestures.
4. Detecting wrist rotation would be better accomplished with a multi-axis accelerometer that includes both a compass and incline function. It would be ideally mounted at the same point as sensor T. This sensor may alleviate the need for the M -> N measurement.









Primitive Measuring Device (Sample 1)

Useful until we have a glove or other automated measuring device.



When fingers would block measurement due to closed against fist measured from top of finger and deducted 10 for finger thickness.













Measurements for sample 1

Measure	A-T	B-T	C-T	D-T	E-T	M-N	T-N	
Calib	103	122	135	122	78	52	83	
B	103	122	135	122	57	52	83	
I	103	0	6	22	52	52	83	
G	8	0	6	22	78	52	83	
H	8	0	135	122	46	52	83	
R	52	47	126	112	56	52	83	
J	103	0	6	22	52	43	78	
Yes	8	0	6	22	52	52	78	
Converted To Ratio of Calibration								
Calib	1.000	1.000	1.000	1.000	1.000	1.000	1.000	
B	1.000	1.000	1.000	1.000	0.731	1.000	1.000	
I	1.000	0.000	0.044	0.180	0.667	1.000	1.000	
G	0.078	0.000	0.044	0.180	1.000	1.000	1.000	
H	0.078	0.000	1.000	1.000	0.590	1.000	1.000	
R	0.505	0.385	0.933	0.918	0.718	1.000	1.000	
J	1.000	0.000	0.044	0.180	0.667	0.827	0.940	
Yes	0.078	0.000	0.044	0.180	0.667	1.000	0.940	

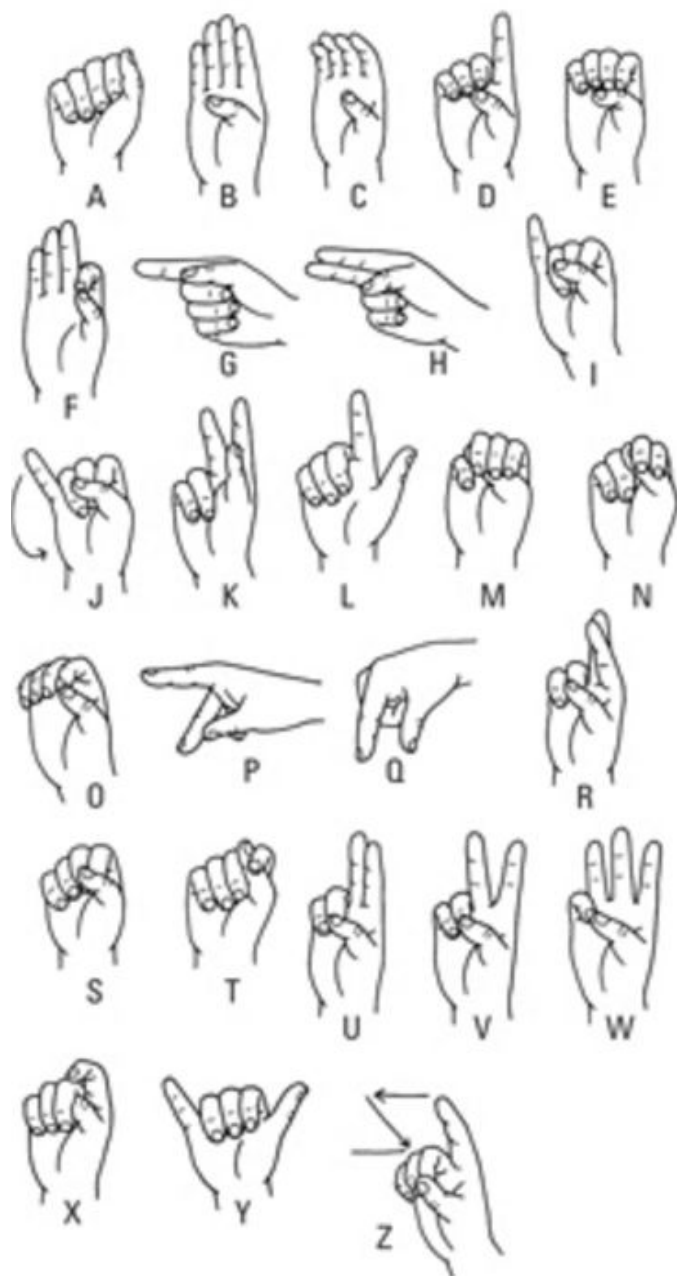
Primitive Measuring Device (Sample 2)



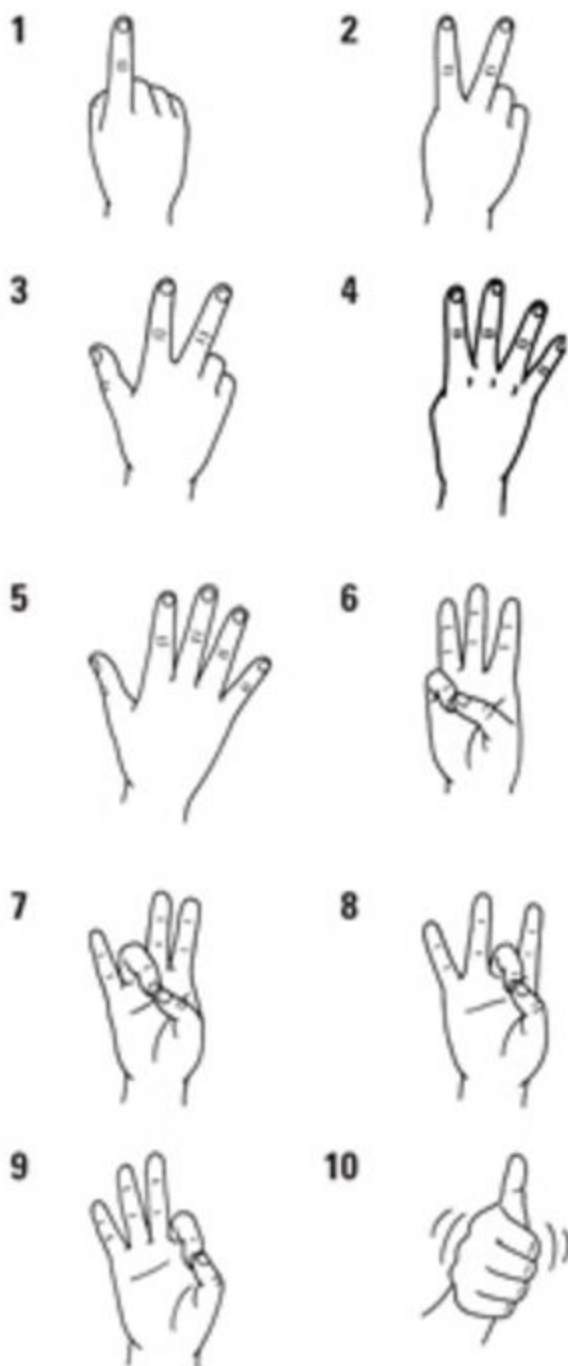
Measures for Sample 2

Measure	A-T	B-T	C-T	D-T	E-T	M-N	T-N	 
Calib	100	122	139	122	76	40	64	 
B	100	122	139	122	46	26	64	
I	100	0	8	20	61	23	60	
G	13	0	13	122	48	33	50	
H	8	0	134	122	48	33	50	
R	47	52	75	88	52	50	50	
J	70	0	17	30	31	61	50	
Yes	4	0	20	25	57	29	49	
Converted To Ratio of Calibration								
Calib	1.000	1.000	1.000	1.000	1.000	1.000	1.000	
B	1.000	1.000	1.000	1.000	0.605	0.650	1.000	
I	1.000	0.000	0.058	0.164	0.803	0.575	0.938	
G	0.130	0.000	0.094	1.000	0.632	0.825	0.781	
H	0.080	0.000	0.964	1.000	0.632	0.825	0.781	
R	0.470	0.426	0.540	0.721	0.684	1.250	0.781	
J	0.700	0.000	0.122	0.246	0.408	1.525	0.781	
Yes	0.040	0.000	0.144	0.205	0.750	0.725	0.766	

Basic Sign Letters



To spell the same letter duplicated use a small bounce between letters or slide the repeated letter over slightly.



Palm toward signer for 1..5.

Palm away from signer for 6 .. 10

HELLO**GOODBYE****NICE TO MEET YOU****YES****NO****PLEASE****THANKS**

Possible Hardware design

Adapt Existing Sensors

The assumption is that we would use a common glove such as VGM light and even though it's measurements are different than those specified here these measurements could be deduced from the available sensors in the glove. The actual glove measurements could be converted to simplify gesture measurements via a staged conversion analyzer as described in our VR Gestures architectural proposal.

Ultra sonic Glove

A simplified sensor glove could be built using a [ultrasonic emission](#) from each measurement point with microphones embedded at points T and U measuring the distance. By using a separate emission pulse from each finger sensor controlled by a micro controller each finger's position can be measured in isolation. Ultrasonic emitters are inexpensive, relatively low power and the sensors are easy to calibrate. They can be built to accommodate multiple gloves in proximity by encoding a id into the pulse which is ignored by other gloves. A similar approach using very low power RF pulses could also be used. It does require adding the emitter to each finger. The approach could be reversed adding the emitter at point T and U and the receiver at the measurement points. Hall sensors were considered but it is ultrasound seems less expensive and lower power than generating a magnetic pulse of sufficient strength.

Keeping the number of emitters low allows the lowest possible duty cycle which will allow more devices to operate in local proximity with less interference. In any case devices powering up should listen for other devices above a given signal strength and time their pulses to take place in the gaps of the devices already operating in the area. Keeping pulse strength as weak as possible will help accommodate more local devices but since ambient noise will have an effect it may be necessary to auto tune pulse strength to the minimum that allows reliable detection.

Sound wave detection transform

Many projects are reading sound using high speed ADC which gives them magnitude. This has a problem of noise and consuming more power than desired. A preferred strategy is to use an [analog boost circuit](#) which is then fed into an analog comparator which we can then time input using a CPU interrupt. Most CPUs only offer 1 or two built in analog comparators but 8 channel chips are relatively inexpensive. This will lose low volumes but it allows the CPU to stay in sleep mode most of the time.

The actual sound detection can be done using either a Fourier Transform, a [Goertzel algorithm](#) which is better for single tone recognition. Another option is [wave similarity analysis from DSP](#). Of these options the DSP approach seems preferable. The circuit using the analog comparator essentially produces a series of on pulses when sound above the threshold occurs with pin at off state the rest of the time. This is measured in the CPU by measuring the arrival times of these On pulses and ambient noise may cause some spurious on states. One of the easiest ways to detect the signal is to look for multiple pulses transitioning from off to on state that arrive exactly the expected amount of time apart. At 50 kHz the pulses would arrive exactly 0.02ms apart so a detectable pulse composed of 5 cycles would consume 0.1ms. If the CPU knows when the group of 5 pulses was set then it can subtract the combined pulse length from the last pulse that arrived and compute transit time.

There are two main ways to encode information in the sound signals which we need to do to allow discrimination between signals when multiple devices are in close operation. The first is to transmit multiple tones simultaneously that are decoded on the receiver side. The other is essentially to modulate the signal into a serial data stream that can be interpreted on the other end. My current preference is using a single modulated tone.

Ultimately the sensor emitter will need a calibration step that allows the emitter to start at low power increasing it's output until the all the receivers are receiving a base tone. Without this step we will get too much cross talk. This should be automated and adaptive to accommodate changing noise levels.

Even with adaptive signal strength control we will still have some risk of sensor cross talk. To minimize data lost due to overlapping signals the basic system should analyze other signals already in use and transmit during the dead time space unused by the other systems.

Sound travels 343 meters per second. It is estimated that our measurement resolution needs to be within $\frac{1}{4}$ inch or 0.00635 meters. Sound will require 0.01840579 ms to travel $\frac{1}{4}$ inch. To be accurate we need to be able to measure differences in pulse arrival at a resolution smaller than 0.0184ms. When using a 20kHz tone and assuming we need at least 5 cycles of signal presence / absence for detection each cycle at 20kHz requires 0.05 ms so it will require 0.25ms to recognize the tone. We can however state reliably that detection will always require 0.25ms so this can be deducted from the time when the pulse is detected. If the high frequency clock counter is running at 16MHz we can discriminate times down to 0.0000625 ms which is 294 times the speed we need to measure $\frac{1}{4}$ " of movement. Under perfect conditions this could allow measurement granularity as small as 0.00129 inch or 775 DPI.

When using a 20kHz tone to transmit data it is easiest to use or 5 cycle detection time measuring presence as a 1 or absence as a 0. This gives us a maximum bit transmission rate of 4,000 bits or 500 bytes per second. Assuming that we are transmitting a 16 bit identifier this limits our measurement rate to 250 samples per second. Assuming that we need to measure at 15 frames per second this would allow up to 16 devices to coexist in overlapping signal space by sharing time slices. This assumes no lost data frames. Higher speeds are available using more sophisticated techniques similar to those used by high speed modems but this is likely to be adequate if we can keep our signal strength moderated to travel no more than a few meters.

By using two emitters and measuring the separate time arrival at the sensors we can increase the sensor precision accuracy by using triangulation and the difference in time of arrival for the signal. Depending on position of the emitters this can provide a full 3D position.

When using a wired or RF connection from the sensor to the CPU we may skip the ID telling the sensor when to expect a signal. The sensor only has to be activated during a short time frame and only needs to the 5 pulses at frequency. Using 5 cycles at frequency at 20 kHz each reading would allow 4,000 detectable pulse emissions per second reduced to 3,000 to allow for de-collision spacing. Assuming each device is emitting 3 discrete pulses per cycle that would allow up to 1,000 discrete measurement cycles to exist in the space where signals overlap. Assuming a frame reading rate of 15 FPS this would give us a maximum of 66 devices co-existing in the same signal space. This would probably be adequate provided the signal strength is kept as low possible for the ambient noise environment. Note high noise environments will have more false sound waves not part of the signals pulses which would cause pulse detection failure and require retry. Using this approach would require incorporating a mechanism where the local CPU monitors pulses unused and computes a open time slice where it can safely send pulses.

We can improve the bit transmission rate and increase the number of co-existing devices by moving to a 50kHz signal but it will require special sensors and speakers to operate at that range. This would also move the system out of the range of hearing for dogs.

A primary weakness in this design is that ambient noise especially high frequency noise could generate so many false input pulses that we can not obtain an accurate 5 pulse cycle. If using 20 kHz with a 5 pulse cycle any noise with a frequency 4kHz or above will conflict. If using a 50 kHz cycle any noise frequency above 11 kHz will interfere. There are many ways to compensate but filtering for 5 ons that are exactly 1/20,000 second apart while ignoring any that do not fit within 5% of the expected arrival frequency is likely the easiest to implement but it does require the ability to handle the case when a noise pulse arrives first or the ability

to retry when that occurs. As such the system will need to keep the timing of all pulses that occur from the time signal transmission begins until it receives the 2 pulse which is exactly the right time after the first one it has the 2nd pulse identified we can use that timing for pulses 3,4,5. If done correctly this could technically allow multiple systems running with the same frequency to operate provided they start transmission at different times. This would favor a sound wave that looks more like a square wave with very short on times over a true sine wave. If we are using square wave type sound emission then we can also filter on pulses that remain on for a length of time different than expected.

Accelerometer added to distance

If cost and power constraints allow then adding a multi-axis including tilt measurement accelerometer to the distance approach would provide adequate data to support more complex moving gestures and to better support gestures such as numbers where the rotation of the hand towards and away from the body have meaning or symbols such as I where movement of the sensor mounted on the pinky is what makes it unique from another gesture or the symbol for yes which requires detecting movement of the hand by rotating up and down at the wrist.

Wireless Sensors using Inductive Sensor Charge

In an ideal scenario we could produce a very small sensor to measure time that could be charged wireless and transmit the timing back to the controller. By adding the ID of the emitter in the measurement transmitted it would allow a unique ability to arbitrarily add additional sensors provided they are close enough to receive exchange data and be charged. The ability to add sensors without wires would make it easier to measure the appendage position relative to head, body, etc. By packaging the sensor transmitter in a very small unit they are ideal for production using very small components at low size and low cost. This approach would eliminate most wires in the product and improve product reliability while reducing cost. The data could be transmitted wirelessly using sound or RF. This is similar to techniques used by [passive RFID systems today](#).

The availability of small ultra low power CPU with instant on such as the TI MSP430 FR series makes it possible to embed the CPU with the microphone sensors. This could make it possible to transmit a pulse of power adequate to charge a local capacitor with enough energy for the sensor and CPU to take the timing measurement and transmit the result back to the main CPU.

The main limit in this application would be the distance we can transmit sufficient power using the inductive pulse. It may also be feasible to produce this charge from motion using pyroelectric, body thermal delta or using a very small battery. The easiest mechanism would use a very small battery or capacitor that can be recharged from an inductive charging station.

RF Pulse Alternative to Ultrasonic

Measuring the time a RF pulse to travel from emitter to sensor could work instead of using ultrasonic sound. The primary advantages of RF pulse devices would be greater resilience to ambient noise. Faster pulses that provide greater idle time space to avoid device contention. The primary disadvantage is that RF moves much faster than sound and the distances we are measuring is small so measuring small changes is more difficult and requires faster CPU which is contradicted by application likely to be battery powered. If using RF it may be more effective to use a measurement of field strength from two angles and triangulate the exact position. The problem with that is that the human body is partially conductive and partially absorptive at RF wave lengths which would make calibration difficult.

Accelerometer based Sensors

It is possible to predict the position of a given sensor if the starting position is known while all movement vectors have been applied. This is used by inertial guidance systems and more recently as a way to augment location precision in GPS applications. The advantage of this approach is all sensors are independent and there would be low risk of cross talk conflicts between sensors in congested environments. The sensors also inherently provide movement, acceleration and rotation data that can make multi-frame classification easier. The disadvantage is any erroneous data can result in incorrect computation of position so there must be provision for returning to a calibration position. Given the noise observed from current sensors this approach while yielding high dividends is likely to require considerably longer to reach a production caliber device.

Stereo Video Analysis

The most useful representation may be using libraries such as OpenCV combined with a sufficient number of cameras to deduce the relative finger positions from the video streams.

One issue with ultrasonic measurement is if the emitter or sensor is blocked when some fingers in a hand are in a closed fist shape. One way to work around this may be to mount the microphone on the back of the fingers. Another could be to add at least a small air gap spacer so the fingers can not close completely to the fist. RF may work better under this condition but very low power RF can also be blocked by the human body.

The Deep learning engines are already supporting video frame classification and are starting to support multi-frame classification after reducing video frames into a series of

vectors. They are still quite limited doing gross classification. We would need to recognize:

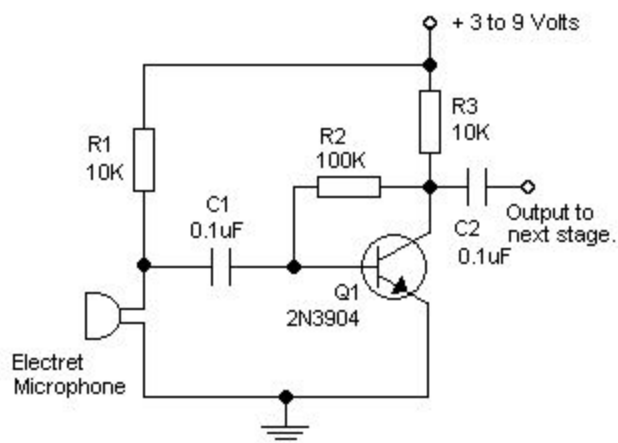
1. Ability to separate one human from the image.
2. Ability to track that human as they move spatially in 3 dimensions through the images.
3. Ability to combine multiple input images to accommodate person and body part rotation.
4. Recognize many body parts such as hands, torsos and fingers
5. Compute relative positions of those body parts relative to each other.

It is clear that Deep learning will eventually be able to support this capability but it is unclear how many man years of investment will be required before recognition can happen at the resolution, accuracy and speed to support widespread ASL gesture recognition in real world lighting conditions. Supporting broader gesture recognition such as football referee movements or american military sign would be easier than ASL.

Reference

- [Goertzel algorithm](#) [Goertzel Library in CPP for Tone Recognition](#)
- [Reliable frequency Detection using DSP Techniques instructable](#)
- [Arduino Frequency Detection: How do you detect frequencies reliably in a noisy signal](#)
Uses a signal shape similarity wave from DSP. Very clever [Autocorrelation, also known as serial correlation](#) to produce [Pitch Detection algorithms](#)
- [Measuring distance with hall effect sensors](#)
- [Measuring distance with sound](#)
- [Development of a glove with Fingertip magnetic sensors](#)
- [ACQUISITION OF VIOLIN INSTRUMENTAL GESTURES USING A COMMERCIAL EMF TRACKING DEVICE](#)
- [Measuring position of fingers for musical instruments](#)
- [Design Considerations for sensor gloves](#)
- [Measuring Distance with parallax](#)
- [Accuracy of ultrasonic measurement](#)
- [Very small microphone mems sensors for distance detectors](#)
- [Various Sign Examples / Dictionary](#)
- [ASL Finger Spell](#)
- [RF ID Basics](#)
- NFC [Magnetic coupling RFID good for upto 10 cm](#)
- [Resonance wireless charging versus inductive](#) Single 2' X 2' charger claims to provide power for an entire room of sensors.
- [Using Accelerometers to Estimate Position and Velocity](#)
- [Measuring distance between two nodes in a Ultrasonic sensor network](#) Kalman Filter
- [Motion Measurement Using Inertial Sensors, Ultrasonic Sensors, and Magnetometers With Extended Kalman Filter for Data Fusion](#)
- [Filtering Sensor Data with a Kalman Filter](#) includes examples in C
- [Multichannel Ultrasonic Data Communications in Air Using Range-Dependent Modulation Schemes.](#)

- [Quietnet a near ultrasonic data transmission using python](#) also a javascript version uses *continuous fourier transform*
- [ULTRASONIC DATA TRANSMISSION WITH GNU RADIO](#) transmitting at 23kHz using FSK free source. Lots of good related links See also [Ultrasound data transmission via laptop](#)
- [Method of ultrasonic data communication and apparatus for carrying out the method US 4045767 A](#)
- [Ultrasonic Local Area Communication](#) Simple but good introduction to the theory [good target frequency is 50kHz](#)
- [Short-Range Ultrasonic Communications in Air Using Quadrature Modulation](#) communications over distances of several meters, using frequencies in the 200 to 400 kHz range.
- [Google use of ultrasound data transmission](#)
- [Fast Hartley Transform \(FHT\)](#) and [Arduino FFT Library](#) [How FFT works](#)
- [Arduino Frequency Detection instructable](#) - nice approach [Adurino SimpleAudioFrequencyMeter](#) uses the Max4466 adjustable gain amp. [Frequency Detector Library](#) [Adurino sound sensor light organ](#)
- [Sound localization theory wiki](#)
- [Frequency detector using PIC 8 bit processor](#)
- [Schematic for circuit recognizing different tones to turn on LED](#) [Simple circuit for sound detection using LM386 opAmp](#)
- [Simple signal boost circuit](#)



SIMPLE AUDIO PREAMP

This easy circuit provides good gain to weak audio signals. Use it in front of an RF oscillator to make an RF transmitter that is very sensitive to sound.

- [Ultrasonic bat detector with two stage LM386 boost](#)
- [Frequency to voltage conversion circuit using LM2917](#)

- <http://electronicsproject.org/high-precision-frequency-and-voice-detector/>
- [Tone Detector Circuit with LM567](#) The simplicity is nice since output is high when no tone and low when tone exists.
- [Pitch perfect tone recognizer built around a MSP433 using Goertzel algorithm.](#)
- [Ultrasonic MEMS Sensor SPM0404UD5](#)