Holographic Memory

8 November, 1995

Michael Brundage

University of Washington

O. Abstract

To increase the speed and capacity of computer systems, many researchers have turned to optical solutions: compact discs, fiber-optic cable, optical computers, and now holographic memories. Modern nonlinear optics poses a number of interesting and difficult (applied) mathematical problems, especially in coding theory. I will describe the optics and codes involved in holographic memories and some potential applications of this technology. No prior knowledge of optics or codes will be necessary.

(See also the enclosed papers *Holographic Memories*, *Three-dimensional holographic disks*, *On Reliable Computation with Formal Neurons*, *Optical memory disks in optical information processing*, and *Programmable image associative memory using an optical disk and a photorefractive crystal*)

I. Introduction - Why Optical Computers?

- Speed
- Capacity

Today, you could build an optical computer. It would be bulky, expensive, and not any better than a PC, but you could do it. However, optical computing is improving so much that I'm convinced there will be entirely optical personal computers within the next few years, certainly within our lifetimes. Two of the main reasons why you might want an optical computer instead of an ordinary electronic computer are greater speed and capacity.

I have a friend in the astronomy graduate department at the University of Hawaii. One of his duties during his first year was to convert astronomy images stored on magnetic tape to optical disk. He told me he was converting about five gigabytes per day, and yet some information would be lost because the magnetic tapes were decaying faster than he was converting them. Now, you and I may not individually think we have such vast storage needs – all of the math department user accounts fit into just a dozen gigabytes – but hospitals and governments certainly have a need to store that much information, and to sort through it quickly. Holographic memories can provide that capability.

One of the most important components in any computer is its memory – the faster and bigger, the better. Compact discs, for example, are very popular today and capable of holding several books worth of information each. Soon a new type of CD is to be unveiled that holds even more data. But even experimental holographic disks hold ten times more than commercial CDs, and are much faster.

II. Basic Holography

- History
- Lasers
- Making Holograms

These days, holograms are everywhere from credit cards to comic-book covers, but it wasn't always so. The first holograms – which weren't very good by our standards – were made by Dennis Gabor in 1947-8 using filtered light. The laser wasn't invented until several years later. It's not entirely coincidental that Hamming, Golay, and Shannon began their explorations into coding theory in 1947-8. Both optics and coding theory have made phenomenal progress in a very short time, and much of that during the 1950's and 60's.

Holograms work differently from the usual compact disc or videodisc, which are in some sense just extremely miniature versions of the phonograph. Some holograms are printed on thin films, which might even be in the shape of a CD, and other holograms are stored in special 3-D crystal substrates such as lithium niobate.

It is difficult to make visible-light holograms – even the tiniest perturbation can produce blurring. However, laser-light holograms are pretty easy to make – there are even isntructions on the Web to make them at home, provided of course that you have a laser.

I made some helium-neon laser light holographs when I was an undergraduate at Caltech. It's a lot loike taking a photograph with delicate (and expensive!) equipment, and then developing it in a dark room. You even use some of the same chemicals used to develop ordinary film. In the end, you have a hologram that, when exposed to laser light, will produce an image of the original objects.

[show slide 2]

The basic idea behind holography is this: Light is a wave. However, ordinary light is travelling every which way, so trying to observe a particular wave is like trying to listen to a conversation at a loud rock concert or a crowded shopping mall. One way around this difficulty is to filter out all the noise until you have light waves travelling in the same direction, all together; another way is to use a laser.

LASER stands for 'Light Amplification by Stimulated Emission of Radiation,' which serves as a succinct description of how it works. A gas, preferrably an inert ones or a mix of inert ones, such as helium and neons, is stimulated by an electric current to give off photons. From physics, we know this will produce light of a particular wavelength. This light is then reflected about and passed through a small semi-transparent aperture, as you see pictured here, and voila'! coherent light.

Coherent light waves can be used to produce interference patterns, which arise from the fact that some waves travelled different distances from the source, and hence are out of phase when they arrive at their destination. You can take advantage of these interference patterns to create or store images of three-dimensional scenes, like the holographs with which we're all familiar.

[describe slide 2]

III. Holographic Memories

- Overview

- Applications

There you have it, holographic memory. Because this process is dependent on the angle of the incoming beams, it turns out that you can store multiple images in a single hologram. You may have seen such holograms, in which the image (and not merely your view of it) changes as you move yourself relative to the image.

These uses mainly regard the substrate as a thin photographic film, but by using its three-dimensionality it becomes possible to store even more images in a single hologram. For example, even a thin holographic disk, less than 1 millimeter thick and the same size as a conventional CD, can have storage densities of about 100 bits per square micron. However, using a 3-D crystal, researchers have stored and retrived about 90 megabytes in a single cubic centimeter. So in other words, a holographic memory of the same size as a compact disk can hold 100 times as much information – and yet this technology is still in its very early stages of development.

Holographic memories exceed the storage capacities of other systems by two orders of magnitude, and yet they are also faster by about two orders of magnitude, having 'seek times' of around 100 microseconds, compared to several milliseconds for the conventional memories. One of the most useful features of holographic memories is the fact that they're associative. Look at this setup here – I've labelled one beam the reference beam and the other the signal beam, but the crystal doesn't know this. To it, both beams are equivalent, so it becomes possible to look up an image by either the reference beam with which it was stored, or by the image itself.

Dmitri Psaltis and others at Caltech used this to build a robot that could navigate their lab. They stored images of the hallways and rooms of the building into its holographic memory, together with instructions on what it should do when it finds that image. They equipped it with a camera, and it moved around the building. You can imagine the FBI looking up fingerprints just by shining their sample into the database, and then checking them against matches that produce strong reference signals in response. Or doing fast searches for patterns in a document. These are things that conventional memories simply cannot do, yet holographic memories do it almost instantaneously.

IV. Basic Coding Theory

- Repetition codes, Hamming distance, etc.
- Linear, BCH Codes

So, holographic memories are amazing things already, and will become even more amazing as research progresses. Researchers have a long way to go before being hindered by any theoretical limits on the capabilities of this technology. However, there are already some significant obstacles requiring mathematics, particularly error-correcting codes. For example, the angle-multiplexing used to store tens or hundreds of thousands of holograms in a cubic centimeter requires an accuracy of about one-thousandth of a degree. Imperfections in the crystal, thermal fluctations in the air, not to mention the reliability of the detectors, lasers, and other equipment involved, all conspire to introduce errors. Even with ordinary compact discs, there are many difficulties such as disk wobble and warp. In practice, the error rates for holographic memories are around 1 bit per ten to a hundred thousand, and by using error-correcting capabilities, researchers have reduced this by a factor of nearly a billion.

[show slide 3]

If you wanted a way to detect and correct errors, how would you do it? For example, suppose you saw this message – what is it supposed to say? Here I've written only twelve of the original eighteen characters, so there was an error rate – actually, an omission rate – of about one-third, and yet you were able to read it. I've heard that in fact English is about 50redundant, so maybe I could have left off another three letters or so.

It's important to realize that there are different kinds of errors. There are erasures, in which the result is garbled; or omissions where it's left out entirely; or errors in which the result is altered to something else. Error may occur with a certain distribution, or they might occur all at once in bursts – for example, a scratch on a CD. Also, there's a difference between detecting that an error has occurred, and actually correcting it. A good code will detect or correct the errors we expect to encounter, without slowing down or bloating our data too much. In particular, codes tend to be application specific.

There is a lot of theory behind all this, but here is a basic introduction to one of the codes popularly used in optics.

A naive version of redundancy could be to repeat each bit to be transmitted some number of times, say three. Now if you receive two one's and a zero, you know that at least one error has occurred, and you could go ahead and decode it to a one since that's the 'closest' possibility. It's possible that two errors occurred, but it's more likely that only one has occurred. We tripled the space needed, but gained the ability to reliably detect and correct up to one error per three bits transmitted. Another scheme would be to append a parity bit to the data. In this example, you can reliably detect up to one error, but not correct it since you don't know in which position it occurred.

For a number of reasons, most research has focused on 'linear codes' – subspaces of a finite dimensional vector space over a finite field, not necessarily GF(2). These linear codes lend themselves to easily described encoding or decoding techniques using basic linear algebra-type manipulations with matrices or polynomials. However, there is a difference between theoretical and practical implementations of encoding/decoding algorithms. Not surprisingly, commercial applications tend to facor those that are practical to implement.

V. Reed-Solomon Codes

- Definition
- Applications to digital audio

One of the most practical codes around are the Reed-Solomon codes. Audio CD's use something called a Cross-Interleaved Reed Solomon Code, that can handle both random errors due to mechanical or material fluctuations and burst errors due to scratches on the disc. In fact, CD players can fully correct burst errors several thosand bits long – which sounds like a lot (and is!) but it really only about 2.5 mm on the CD itself.

[show slide 4]

If a code is not merely a subspace of a vector space, but is in fact cyclic, then there is additional structure to it which makes it even easier to work with. We can let codewords correspond to polynomials over a particular field; in fact, cyclic codes correspond to ideals in $F[x]/(x^n - 1)$ which in turn (from abstract algebra) corrspond to monic polynomial divisors of $x^n - 1$. Cyclic codes have a number of useful properties, especially decoding and encoding algorithms that are easily implemented with digital electronics.

Moreover, it's possible to create cyclic codes, called BCH codes, with designed distance between codewords. This is important, since (as we saw previously), if a code has distance d, then we can correct up to $\lfloor (d-1)/2 \rfloor$ errors. Reed Solomon codes are a special kind of BCH codes. If α is a primitive nth root of unity in $GF(q)$, then the polynomial $(m \geq 0, d \geq 2)$ $f(x) = (x - \alpha^{1+m})(x - \alpha^{2+m}) \ldots (x - \alpha^{d-1+m})$ divides $x^n - 1$ so it generates an $(n, n-d+1)$ Reed-Solomon code over $GF(q)$. Although in a BCH code the actual distance may exceed the designed distance, in a Reed-Solomon code they are equal. Thus an (n, k) RS code can correct $\lfloor (n-k)/2 \rfloor$ errors. Also, an RS code over $GF(2^m)$ can be though of as a code over $GF(2)$ by replacing elements in the larger field with m-tuples of elements in the smaller field. In this way, one obtains an (nm, km) code that can correct burst errors of up to length $m\lfloor (n-k)/2) - 1 \rfloor + 1$.

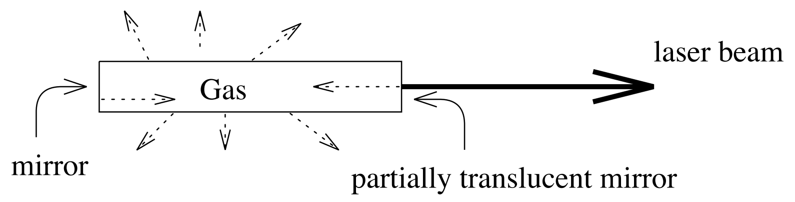
[see example on the slide]

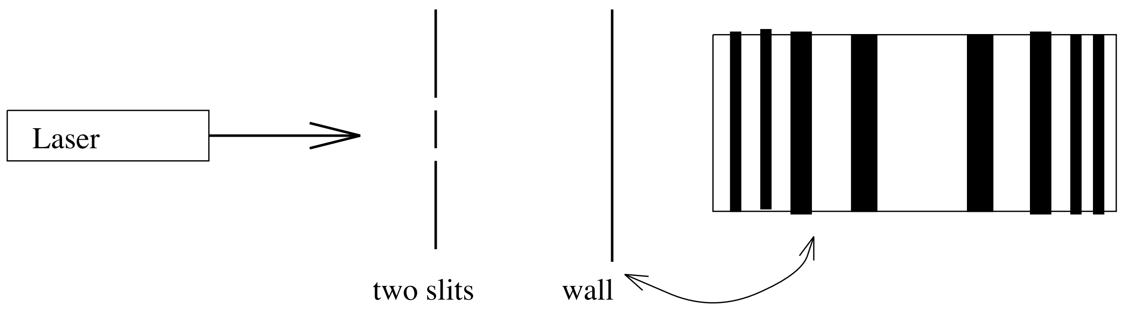
Another trick to improve the burst error-correcting capacity of a code is to interleave it. For example, if one wishes to transmit the three codewords (1100001), (0011110), and (0111000) in that order, one could instead treat them as successive rows of a matrix and then send the columns instead. Now a burst error of length three will really only result in a single error in each codewrod. So if our original (7,4) code could correct burst errors of length 1, now we have a (21,12) code that can correct bursts of length 3. You can also do variations on this trick, such as delayed interleaving to spread out the errors within the codewords, and so on.

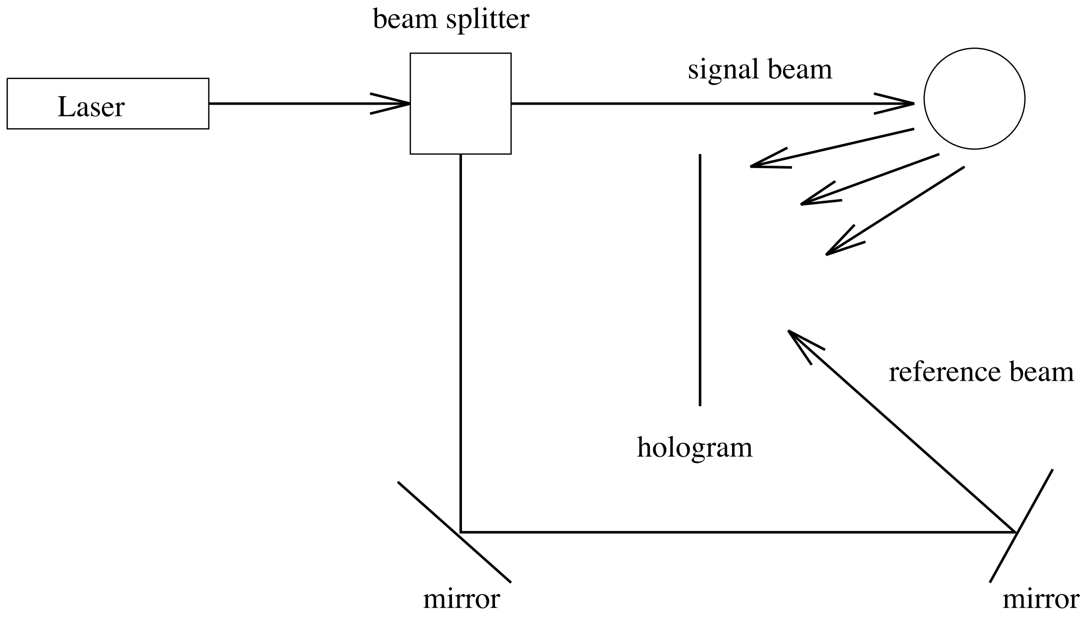
So now we come to the actual codes used in CD players.

[Describe details for digital audio encoding/decoding, EFM, etc.]

Laser = Light Amplification by Stimulated Emission of Radiation







CN U RD THS?

C▯N ▯▯U R▯▯D TH▯S?

CBN X0U RAED THYS?

A simple repetition code with 'majority vote' decoding

$0 \longrightarrow 000$

$1 \longrightarrow 111$

$111 \longrightarrow 1$

$110 \longrightarrow 1$

$101 \longrightarrow 1$

$100 \longrightarrow 0$

A single parity bit encoder and decoder

$11 \longrightarrow 110$

$10 \longrightarrow 101$

$01 \longrightarrow 011$

$00 \longrightarrow 000$

$110 \longrightarrow 11$

$101 \longrightarrow 10$

$111 \longrightarrow ??$