

Emerging Technical Opportunities VII

We seem to be overdue for yet another in our series of emerging technical opportunities. Stuff that is newly positioned so any individual or smaller scale startup can profit from it.

You can see where we are coming from and where we are heading with these previous files...

[Emerging Opportunities I](#)
[Emerging Opportunities II](#)
[Emerging Opportunities III](#)
[Emerging Opportunities IV](#)
[Emerging Opportunities V](#)
[Emerging Opportunities VI](#)

[Resource Bin #63](#)

What I thought I'd do this time around is stay somewhat closer to home. Looking at projects I am actually working on. Or shortly would like to.

We can start with my biggie of...

Steplocked Magic Sinewaves

For decades now, I have been searching for the "perfect" digital power sinewave synthesis technique. Using *elegant simplicity* and improved energy efficiency for everything from home and industrial motor controls through electric vehicles to solar energy to aerospace apps.

The latest of my *steplocked magic sinewave* versions newly offer...

- *simple switching from a dc supply*
- *all low harmonics forced to zero*
- *fewest possible switching events*
- *highest possible efficiency*
- *storage needs as low as seven bytes*
- *optional three phase compatibility*

The use of *Chebyshev Polynomials* in the synthesis process strongly suggests these results are in fact optimum, for the maximum possible number of low harmonics forced arbitrarily close to zero. While using the minimum possible number of switching events.

I've greatly expanded my *Magic Sinewave* library page on my *Guru's Lair* web site. Here you should find lots of *tutorials*, JavaScript *calculators*, and related info.

Your "best" magic sinewaves are often the *Steplock 28's*. A "best efficiency" steplock 28 might use seven pulses per quadrant and force the first 28 harmonics to zero. The fully "delta friendly" alternate can force the first 22 harmonics to

zero. Using as few as *seven* bytes of storage per amplitude! As many as several thousand harmonics can be zeroed out using the currently available development tools.

Partners are welcome to explore this new billion dollar unique energy efficiency opportunity. Start off viewing this *Development Proposal*. Seminars plus *Custom Design Assistance* are also immediately available.

PostScript as Language

Another subject I might have mentioned a time or two before is that superb general purpose PostScript computing language. Few people realize how simple, fun, and easy PostScript has recently become. Let alone the astounding things you can do with it.

Why, it has even been rumored that you can dirty up otherwise clean sheets of paper with it! But there is no point whatsoever limiting yourself to such trivia.

These days, most anybody serious about any web work already owns a copy of *Adobe Acrobat*. The big secret is that host based *Acrobat Distiller* makes a jim dandy host based universal PostScript interpreter. All you do is write a plain old ASCII textfile with *Wordpad* or your favorite WP or editor and send it to Distiller. PS program commands and visual display are but a few seconds apart.

Distiller then generates *four* forms of useful output: A printable, viewable, and web deliverable totally controlled PDF document; your on-screen messages; a log file that can report or save info; or an optional disk file.

Besides being the obvious choice anytime extensive or exotic calculations get mixed with text or graphic display, PostScript also has lots of potent *Video Uses*. Plus largely yet unexplored *Robotics Apps*. But the very best of all is PostScript's little known and poorly appreciated ability to *read and write any disk file in any data format*. For instance, exotic mining of your ISP's log files are easily done using PostScript. As are *bitmap manipulations*.

I use "raw" PostScript for all of my writing and virtually all of my engineering design. Helped along now and then by JavaScript when I absolutely need 64-bit math.

Speaking of which, I routinely use PostScript to *write* my JavaScript programs! A highly customized 4 megabyte *Javascript Calculator* can be *written by PostScript* in four seconds! Done by hand, it was taking me hours.

I've updated and expanded my *PostScript-as-Language* library page. Included are my *Gonzo PS Utilities*, classic and modern tutorials, *Show-and-Tell* and *Secrets* demos, *Beginner Projects*, links, and *books*.

Privatized Military Surplus

There have long been incredible opportunities centered upon buying and reselling military surplus. Electronics and otherwise. First and foremost, as an income tax refund that can easily exceed 500%. Second, to build yourself up a dream lab. Third, for an **eBay** cash cow. And finally, as a low key, low maintenance source of income that funds the stuff you really want to do.

Average prices typically spanned 0.8 cents on the dollar. Sadly, somebody finally seemed to have noticed that the **DRMS** bureaucracy was spending over \$1.60 in admin costs to close each \$1.00 in actual sales. And that dumping their stuff in the driveway or using it for flood control riprap would have been a lot more cost effective.

Hence privatization.

As of October 2001, most useful military surplus sales have gotten privatized. All that remain done the "**old way**" are a few special hazmat, term, and offshore sales.

Instead, a commercial, private, and for-profit venture which goes by the three names of **Levy-Latham**, **Surplus Bid**, or **Government Liquidation** now offer the bulk of the more useful and interesting items.

The good news here is that buying is now ridiculously more convenient, better described, and far better imaged. With fewer bureaucratic hassles, bungling, and surprises. The bad news, of course, is that the prices have gone up somewhat and are virtually certain to further increase in the future. On the average, though, the process is now way, way better than before.

Some of my own surplus offerings appear on our **eBay** auctions and on our web **Bargain Pages**. Ongoing details of **Equipment Refurb** and my **My eBay Secrets** tutorials plus lots more are on our **Auction Help** pages.

Intelligent Web Technology

It should go without saying that a decently done website is crucial for any tech venture. Your site should give your users unique sole source content, besides being a definitive link gateway to everything else in the field. Leading with your personal value added and giving a lot away.

The key question is: What technologies do you exploit and which do you ignore? What is essential, and what is bandwidth wasting glitz? Let us take a brief look at what seems to be working for me.

For openers, all of your technical content **must** be in the **Adobe Acrobat** PDF format. First to give you total control of exactly what your user should see. Second, to insure the highest possible quality. Third for stuff like being easy to magnify or getting read to a disadvantaged user. Finally, for Acrobat's being the industry standard format for just about all printing and publishing.

Acrobat PDF files can also be optimized for **Byte Range Delivery**, which means that early pages can get viewed as later ones download in the background.

There are several non-obvious things you can do with Acrobat that greatly improve its web presence. Naturally, your web site **must** have a local and private search engine. Adobe has an **IIS plug in** that will extend your normal search to include all words in all **Acrobat** PDF files.

It is a fairly simple matter to add bells and whistles to PDF docs to make them look and behave in the manner expected of web HTML files. **Buttons and Tables** can be

easily added with this tutorial. Links that look and behave like HTML are easily created. Either manually with Acrobat itself, or by using this **fully automated linking example**. Yes, this code automatically moves link positions around on the page as edited text changes.

"Thumbnail" **slide shows** are easily done using Acrobat. With the key advantage of being a single quick loading file instead of fragmented all over the place. Combined with the total control of exactly what your viewer should see. As **this sourcecode** shows us, it is also a simple manner to merge existing JPEG files into a PDF doc.

Dynamic web pages seem the rage these days. Whose content is changed on request or alters with user actions such as a mouseover. Some of this is best done at the client end using JavaScript routines. Other tasks are best handled at the server end. While server-side tricks traditionally were done with CGI, a newer approach uses ASP.

Short for **Active Server Pages**.

The two biggest needs that I've had for ASP are **banner rotation** and **includes**. Includes are simply subroutines you enter once that will magically reappear on each page. Such as headers, footers, nav menus, or file lists.

While there are stock banner rotators available, I wanted one that would randomly place eight ads per page, each one showing up only once with ever changing positioning. As you can see in this **Banner Rotation** tutorial and this **Banner Advertiser** page.

By the way, always be sure that your **banners load last!** Creative use of LOWSRC is the secret. Give your viewer something to view as quickly as possible.

Over on the client side, I often like to use some simple JavaScript routines to spruce up my page appearance and actions. My **bouncy buttons** behave in pretty much the same way as any other mouseover does. But my **bouncy bricks** are unique, though, and work best in IE. A bouncy brick is really nothing but a table cell whose background color changes on any mouseover. Besides being a subtle addition, they dramatically shorten and simplify any code that involves dozens or hundreds of menu selections.

All the tiny details really matter in any decent website. Such as the use of Meta Tags to make your pages way more searchable and better identified. Pay special attention to your **mid-level** nav, adding a layer that simplifies a user moving around or between pages. Make it easy for them to return to where you want them to go. Link every offering to at least one other.

Test. Test. Test.

Then test some more. Use **Dr. HTML** or a similar website testing page, combined with all your usual spell checkers. Carefully monitor your log files to find out when and how your users are getting stuck. **Webtrends**, while somewhat pricey, can be tremendously useful here. Make your **Daily Stats** and **Monthly Stats** available on your website.

Both for your own use and others.

Log files are easily mined for an incredible stash of info. I've written a few **PostScript** utilities that let you pull out the hidden goodies. Such as discovering when someone rips off your eBay images. Or separating auction hits from referrals. Or link list extraction for PDF file checking.

Finally, a secure payment page is a must. Allowing direct purchases with VISA or MasterCard. I'm still exploring this, so I don't have specific recommendations at present.

SOME RECOMMENDED SVG BOOKS

Jacek Artymiak, [Data Visualization on the Web with SVG](#)
Andrew Watt, [Designing SVG Web Graphics](#)
Jon Ferraiolo, [Scalable Vector Graphics \(SVG\) 1.0 Spec](#)
David Eisenberg, [SVG Essentials](#)
Micah Laaker, [Teach Yourself Svg in 24 Hours](#)

For more book info: www.tinaja.com/amlink01.html

Examples of these techniques are found on my [Guru's Lair](#) web site. More specific support on these topics is in my [BOUNCY.PDF](#) or [BWHISTLE.PDF](#) tutorials. Viewed on my [Acrobat](#), [Blatant Opportunist](#), [PostScript](#), [Fonts](#), and [Webmastering](#) pages. Custom consulting and design is available per our [InfoPack](#) services.

SVG Scalable Vector Graphics

What will be the "next big thing" on the web?

There's now a move afoot to make a non-Adobe version of something like Acrobat PDF a built-in part of the web. Giving you the short files and absolute appearance control of Acrobat without needing any special plugins.

It is called **SVG**, short for **Scalable Vector Graphics**.

Unlike previous HTML, SVG supports text, images, and vector graphics shapes. These shapes can be grouped as subroutines, be clipped, transformed, extended, templated, or alpha masked.

SVG 1.0 is now a web standard, and we are now at the "chicken and egg" stage of getting browser support and a popular enough user base. I see nothing preventing this from taking off big time.

For more info, you can start off with this [SVG Overview](#) from [W3.ORG](#), the web standards folks. Then pick up the new development tools from [Adobe Systems](#) and others. Adobe has free SVG viewers available. Finally, check out the [SVG book listings](#) shown in the sidebar.

Playing with Pixels

In these days of two gig processors, it is a trivial matter to take an image and rework each and every pixel in as fancy a manner as you care to. So it would seem that there are now all sorts of new image manipulation opportunities opening up. Let us look into a few image manipulation concepts that I've been exploring...

upgrading product photos— I've always been very big on maximizing personal value added as a success secret. I have found that spending considerable extra time on my [eBay](#) and [Bargain Page](#) images easily doubles prices.

While [Adobe Photoshop](#) is often used here, I've gotten by just fine by using plain old Paint combined with the [ImgViewer/32](#) shareware from [Arcata Pet](#).

For additional details, check out my [MYEBAYS.PDF](#) or [STPPREP.PDF](#) and similar files on our [Auction Help](#) library page on my [Guru's Lair](#) website.

reading the fine print— Ordinary fonts tend to end up useless in the tiny sizes needed for image retouching or bitmap manipulations. One solution is to create your own custom single sized **bitmap fonts**. A better one is to make up **true antialiased** mini fonts. With true antialiasing, each font pixel gets replaced by a gray (or proportional

color) value equal to the average character energy in its allotted space. As [this collection](#) shows you, amazing legibility can result.

These fonts are often used with [PostScript-as-language](#) to directly create Bitmaps you can copy and paste into Paint. [This Demo](#) shows us a few possibilities; [sourcecode](#) and many more tools are found in our [Font & Bitmap Manipulation library](#).

faking alpha channels— Ordinary PostScript does not normally permit alpha blends. In which any fractional opacity of an overlay can appear. But this [Alpha Blender](#) demo shows us how alpha channels can be effectively faked. Albeit somewhat awkwardly. The trick is to create two images and then selectively blend each and every pixel by a chosen ratio. The use of PostScript's obscure **infill** and **instroke** operators can be handy here.

More details are available via our [custom consulting](#).

combining topops and images— The web topo services such as [Terraserver](#) or [Topozone](#) are still missing a key feature—the ability to selectively merge topo maps and aerial photography in any blended ratio.

This combined topo+aerial image of [PacMan, Arizona](#) shows us how its done. All you really have to do is alpha blend and then bump the contrast. Per [this sourcecode](#).

color keying— But since we can now custom calculate each and every pixel, you can do much, much more than simple blending. For instance, topo contours are tan or brown. You can pull out only the contours by selectively looking for certain color ranges. Thus different features can be extracted by their colors or intensities.

This, of course, is just an extension of the weatherman's blue screen you see on tv. But we have newly the ability to mask extract info on hue or brightness. For exceptional versatility. More on our [Fonts & Bitmaps](#) library page.

more photo magic— One feature I sorely miss from back in my view camera days is swings and tilts. By grossly distorting a view camera, you could do such tricks as architectural 3-D perspective where all vertical lines stay that way. Or align things that are supposed to be truly square through **image rectification**.

These days, we can remap pixels instead, creating view camera effects without the muss and fuss.

Another world of interesting possibilities is to create a **shading mask**. This can be used for classic dodging and burning. Otherwise known as selective brightening or darkening. Or doing vignetting. Where outside edges can be feathered or have their contrast altered.

Again, [consulting services](#) are available on any and all of these topics. There's some real exciting possibilities here.









[Let's Hear](#) from you. ♦

Microcomputer pioneer and guru Don Lancaster is the author of 35 books and countless articles. Don maintains a US technical helpline you will find at (928) 428-4073, besides offering his own [books](#), reprints and [consulting services](#).

Don also offers surplus bargains found on [eBay](#) and on his [Bargain Pages](#).

Don is also the webmaster of www.tinaja.com You can also reach Don at Synergetics, Box 809, Thatcher, AZ 85552. Or you can use email via don@tinaja.com

PLEASE CLICK HERE TO...

-  Go to the [main library](#)
-  Start your [tech venture](#)
-  Sponsor a display [banner](#)
-  Find [research solutions](#)
-  Send Don Lancaster [email](#)
-  Pick up surplus [bargains](#)
-  Find out what a [tinaja](#) is
-  View recommended [books](#)