An Improved Bitmap Image Keystone Corrector

Don Lancaster
Synergetics, Box 809, Thatcher, AZ 85552
copyright c2005 as GuruGram #55
http://www.tinaja.com
don@tinaja.com
(928) 428-4073

One popular drawing and imaging format is called **two point perspective**. And is otherwise known as "**architect's perspective**". In which there are two more or less eye level vanishing points to your far left and far right. And where **all your vertical lines remain that way**.

This **keystone correction** distortion has long been used to make all building sides (and telephone poles in particular) vertical in street scenes and for construction renderings. I've found a similar technique to dramatically improve the viewability and final pricing of our **eBay** offerings...

Such techniques can also be used in reverse to flatten an image area and convert it from perspective to "flat on" rectangular. Lettering and graphics improvement can then be more easily made. Followed by an "unflattening" back to an original trapezodial form. We saw details on this in **PERSPEC1.PDF** of **GuruGram #54**.

The traditional keystone correction got done using the **swings and tilts** on a view camera. Lesser cameras could pretty much fake the same effect by tilting the table on an enlarger. I've previously done a keystone corrector as **SWINGTLT.PDF** whose underlying **PostScript** utility is **SWINGT01.PSL** and whose tutorial can be found as **GuruGram #15** .

This older code has now been upgraded into a new **FIXTLT01.PSL** keystone corrector. Improvements include a more intuitive and more precise 4-value entry, faster speed, and better linearity. There is now considerably less trial and error. Flattening and reperspecting is simplified. Optional white lockout to prevent any background punchthrough is now internal. The needed key **Gonzo Utilities** have also been internalized, so use of my full **Gonzo Utilities** are now optional.

Keystoning Fundamentals

Either keystoning or keystone correction gets done by reworking **one horizontal line at a time** in your image. Pixels on that line are shifted as needed to the right or the left of their old positions to provide the desired effect. **There is no vertical shifting of any pixels.** The correction is **horizontal only**.

In general, **pixel interpolation** will be needed for best results. This can involve the **high resolution cubic spline interpolation** that we detailed in **GuruGram #4**. Keystone correction nicely interacts with the **.BMP Image Format** which works one line at a time from the image **bottom upwards**.

Strong correction is made at the top, getting progressively weaker till a zero correction point exactly halfway up the image at our **tilt axis**. Negative correction then continues to the bottom, getting progressively stronger.

Two key concepts are the **neutral axis** and the **gain**...

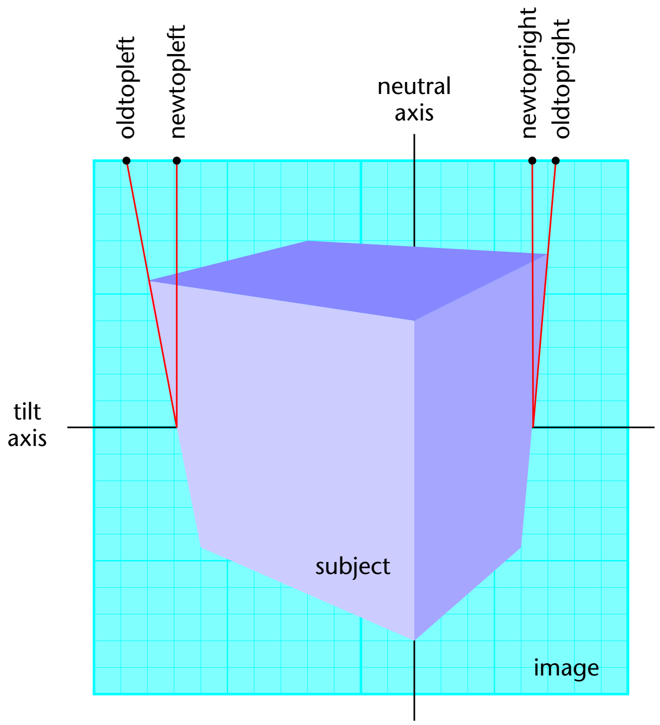
NEUTRAL AXIS - The vertical line of zero correction.

GAIN - Correction proportional to distance from neutral axis.

The neutral axis is thus **a vertical line at which no keystone correction will take place**. Note that the neutral axis is only in the "middle" when **symmetric** repairs are needed. In general, the neutral axis can be anywhere on or off of the image.

A positive gain moves you **away from** the neutral axis, while a negative gain moves **towards** the neutral axis. A **unity gain** makes no change in pixel position.

Four data entry points are needed along the **top line** of your image. In our new **FIXTLT01.PSL** , these are **oldtopleft**, **newtopleft**, **oldtopright**, and **newtopleft**. These values are easily determined by using the selection rectangles in **Paint**, in **ImageView32**, or other image manipulation programs...



Note in particular that these data points are the **vector extensions** of the subject lines and **not** those of the subject top itself.

Some Code Excerpts

As with most of my utilities, I use **PostScript** as a **General Purpose Computing Language**. Adobe's **Acrobat Distiller** is especially adept when serving as a **host based PostScript interpreter**. This scheme is quite useful when creating most any disk based file in virtually any format. Three outputs are possible: The usual .PDF file which we do not use here, the log file to return useful info, or new disk files written to disk in **any** format.

In general, you bring a copy of **FIXTLT01.PSL** by using most any editor or wp program, change your four data points, edit your filename, and decide if you want the optional white punchthrough removal active. This **standard ASCII text file** is than saved under a new name and sent to **Acrobat Distiller**. A new and corrected **.BMP bitmap** file will then be automagically generated.

A file auto renaming gets done by changing the **sixth** filename letter from the right to a lower case **"k"** for keystone. A **mypix01.bmp** outputs as **mypixk1.bmp**.

Code processing begins by finding the left and right differences or **deltas**...

```
/leftdelta newtopleft oldtopleft sub store

/rightdelta newtopright oldtopright sub store
```

These are "backwards" because of a hidden gotcha: **.BMP Bitmaps build from the bottom up!** Our code will really begin on the **bottommost** line instead of the top line where the shift points have been more conveniently entered.

The neutral axis and the gain are then readily computed...

```
/neutralaxis newtopright newtopleft sub leftdelta dup
      rightdelta neg add div  mul newtopleft add store

/gain leftdelta neutralaxis newtopleft sub div store
```

The amount of keystone shift is a maximum on the top line, is zero on the center line, and a maximum negative on the bottom line...

```
/vgfract curvline vres 1 sub div 1 sub neg 2 mul 1 sub store
```

Each successive line pixel gain is scaled by **vgfract** as needed.

The program **reads** one diskfile image line at a time, **modifies** that line by repositioning its pixels, and then **writes** that line to a new diskfile. There is less disk clatter and considerble speedup by working a full image line at a time.

The secrets behind a decent pixel interpolation are found in **BASIS.PDF**. To interpolate a pixel, **four** adjacent pixels are scaled by high resolution cubic spline table lookup values and added together.

Punchthru Elimination

NEWBKG01.PSL is an example of an automatic backgrounding and optional vignetting utility that replaces each white background pixel with a computer new background value. A tutorial of **a previous version** appears as **GuruGram #17** .

Besides dramatically improving appearance, such autobackground routines can also greatly minimize JPEG edge artifacts.

Annoying to distastrous results can happen if there are also white pixels **inside** your active image areas. This can create **punchthru** which puts background pixels in the middle of your active image.

And is clearly ungood.

Punchthru elimination is easily included in a keystone corrector as an option. A cubic spline pixel interpolator can sometimes end up with "whiter than white" or "blacker than black" pixels which have to be trapped out and substituted. If the white substitution is a **255**, a normal white is replaced. If the white substitution is a **254** instead, a fake white results. A white that looks like a real one but does not punch through.

Perspective Lettering Correction

We saw some sneaky tricks that let you dramatically improve perspective lettering on bitmaps in **PERSPEC1.PDF** of **GurGram #54** . This requires a vertical **swing** correction rather than the horizontal **tilt** adjustment we have been looking at.

Swings (and true XY image rectification in general) tends towards more complex and slower code. So your best route to a swing is often to rotate your image 90 degrees, do a tilt or keystone correction, and then rotate 90 degrees back. It is important that you **do not crop** midway in this process!

For More Help

Additional info on image manipulation for **eBay** use is found on our **Auction Help** library page. More on **PostScript** and **Acrobat** in their separate resource areas. More on bitmaps in our **Fonts & Images** library. Free **Gonzo Utilities** and many use examples are found **here**.

Additional consulting services are available per our **Infopack** services and on a contract or an hourly basis. Additional **GuruGrams** are found **here**.

Further **GuruGrams** await your ongoing support as a **Synergetics Partner**.