

Comparing Two-Dimensional Electrophoretic Gels Images across the Internet

[Peter F. Lemkin](#)

Image Processing Section / [LECB](#)

Bld 469 Room 150

NCI-FCRDC/NIH, Frederick, MD 21702, USA

Keywords: image comparison, World Wide Web, Java, Internet, two-dimensional electrophoresis, databases, proteins/genetics, human, electrophoresis, gel, two-dimensional, proteins/analysis, image analysis.

Note: This work was presented at the Sept 15-18 1996 conference ``FROM GENOME TO PROTEOME: 2nd Siena 2D electrophoresis meeting" in Siena, Italy.

Original paper: [Lemkin PF \(1997\)](#) Comparing Two-Dimensional gels across the Internet. *Electrophoresis*, **18**, 461-470. (The paper was revised December 10, 1996.)

This extension of the paper was revised (\$Date: 1998/11/20 19:24:08 \$) to reflect recent changes in Flickr.

Also, see [Lemkin PF \(1997\)](#) 2DWG meta-database of 2D electrophoretic gel images on the Internet. *Electrophoresis*, **18**, 2759-2773. [[Extended version of the 2DWG paper](#)]. The 2DWG incorporates Flickr to compare gels from a world database of 2D-PAGE gels with gels from the user's Web site.

ABSTRACT

Scientists around the world often work on similar data so the need to share results and compare data arises periodically. We describe a method of comparing two 2-dimensional (2D) protein gels of similar samples created in different laboratories to help identify or suggest protein spot identification. Now that 2D gels and associated databases frequently appear on the Internet, this opens up the possibility of visually comparing one's own experimental 2D gel image data with data from another gel in a remote Internet database.

In general, there are a few ways to compare images: 1) slide one gel (autoradiograph or stained gel) over the other while back illuminated; or 2) build a 2D gel computer

database from both gels after scanning and analyzing these gels. These are impractical since in the first case the gel from the Internet database is not locally available. In the second, the costs of building a multi-gel database solely to answer the question of whether a spot is the same spot may be excessive if only a single visual comparison is needed.

We describe a distributed gel comparison program (<http://www-lecb.ncifcrf.gov/flicker>) which runs on any World Wide Web (WWW) connected computer and is invoked from a Java-capable web browser. One gel image is read from any Internet 2D gel database (eg. [SWISS-2DPAGE](#)) and the other may reside on the investigator's computer.

Images may be more easily compared by first applying spatial warping or other transforms interactively on the user's computer. First, regions of interest are "landmarked" with several corresponding points in each gel image, then one gel image is warped to the geometry of the other. As the two gels are rapidly alternated, or flickered, in the same window, the user can slide one gel past the other to visually align corresponding spots by matching local morphology. This flicker-comparison technique may be applied to analyzing other types of 1D and 2D biomedical images.

1. INTRODUCTION

The World Wide Web (WWW) or "the web" [1] is a vast set of interconnected documents and databases available over the Internet that has changed how scientists communicate their data. It offers the opportunity to publish and compare data in a way that is immediately available to the scientific community with access to the Internet. Many biomedical web servers have been set up -- too many to enumerate here. These servers themselves contain lists of URLs of other biomedical databases. <http://www-lecb.ncifcrf.gov/EP/table2Ddatabases.html> lists some of these biomedical servers which may be useful as starting points in your searches.

As 2D PAGE gel protein maps have become increasingly detailed, it is becoming easier to try to identify or suggest protein spot identification by comparing one's experimental gel against published 2D gel maps such as SWISS-2DPAGE [2-4] and many others. More spots in 2D gels are being identified [5-6]. For several years many 2D electrophoretic protein gel databases have been published, including some in special 2D gel database issues of *Electrophoresis* [7-10] including a special issue on electrophoresis in cancer research [9]. Obviously, it is easier to compare gels that have been prepared using exactly the same type of apparatus and chemistry. However, it may still be useful to compare gels with similar material even though the gels were run under

somewhat different conditions. SWISS-2DPAGE estimates the range of pI and MW of proteins based on the amino acid sequence as well as the actual values measured in 2D PAGE gels [11]. So there may be some utility in knowing where the spots of interest might appear in a gel. The problem of comparing 2D gels across the Internet was first raised at the 1994 Siena 2D electrophoresis conference on proteins and the genome [12], and this started my thinking about how to use flickering to solve it.

There are a several ways we could compare two 2D gels. If the gels are complex, simple observation may not be adequate. We could manually slide one backlighted autoradiograph or dried stained gel over the other. Alternatively, we could build a 2D gel computer database of matched spot data using scanned images of the two gels. The latter assumes that we have access to both of the gel image files, and have in hand the specific computer and analysis software required to analyze these gels. Many such gel analysis systems have been developed in research environments [13-19] as well as those offered by commercial vendors. Both of these solutions are impractical for most investigators if only a single visual comparison is needed to check one spot. In the first case, the gel corresponding to the Internet database is not locally available (although one might view it with a web browser, save it, and make a transparency at reduced resolution). In the second case, the amount of work required in building a multi-gel computer database of these two gels solely to compare two spots is excessive.

Visually comparing two images to determine if two objects (such as spots) are possibly the same depends on successfully matching the local morphology around the two objects. If it does match, then one has more confidence that the objects are the same. It is this definition of putative identification which we strive towards in our definition of comparison. Full identification requires further work such as cutting out the spots and subjecting them to sequence analysis, mass-spectrometry, testing them with monoclonal antibodies, other methods.

In the method being presented, the investigator must scan their gel(s) onto their local computer that is connected to the WWW. One gel image may be obtained from any Internet 2D gel database while the other is obtained from the investigator's local computer. Alternatively, both images may come from the Internet or both from the local computer. Flickering gel images on a computer is one way to compare gels without requiring physical gels or autoradiographs. Mechanical flickering had been used in the past in astronomy for comparing star maps. N.G. Anderson had used it for comparing 2D gels. We have used the flicker concept for over a decade in our GELLAB 2D gel database analysis software system ([13-14], [20]), as well as in our Internet-based image conferencing system Xconf [21]. Some of the ideas for our use of distributed Internet databases for biomedical images were developed in our Protein Disease Database system [22-24], and in the Mitochondrial Database [25] which use the ``[federated databases](#)'' concept. Appel has nicely described the federated database concept of

creating and sharing distributed databases [26] on the Internet primarily through the World Wide Web. Such federated databases are becoming increasingly popular as means of sharing data without having to independently create, maintain or copy data. This greatly reduces the costs of using the data since the user or their local support does not have to copy or maintain copies of these databases. [Table 1](#) lists the web URL addresses of a number of 2D protein gel databases which might be considered to be part of a 2D gel federated database.

Table 1. Partial list of World Wide Web 2D electrophoretic gel databases. Individual gel images with identified proteins are available in these databases. The user should investigate them individually since URL paths for 2D gel image files will differ. An updated list is available at <http://www-lecb.ncifcrf.gov/EP/table2Ddatabases.html>.

-
1. ExPASy (liver, plasma, CSF, etc., SWISS-PROT, SWISS-2DPAGE),
URL: <http://www.expasy.ch/>
 2. CSH QUEST Protein Database (yeast, REF52 rat, mouse embryo),
URL: <http://siva.cshl.org/>
 3. E.coli Gene-Protein Database Project - ECO2DBASE (in NCBI repository)
URL: <ftp://ncbi.nlm.nih.gov/repository/ECO2DBASE/>
 4. Argonne Protein Mapping Group (mouse liver, human breast cell, etc),
URL: <http://www.anl.gov/CMB/PMG/>
 5. Heart Science Centre, Harefield Hospital (Human Heart 2D gel Protein DB),
URL: <http://www.harefield.nthames.nhs.uk/>
 6. Cambridge 2D PAGE (rat neuronal database),
URL: <http://sunspot.bioc.cam.ac.uk/NEURON.html>
 7. Embryonal Stem Cells (Immunobiology, University of Edinburgh),
URL: <http://www.ed.ac.uk/~nh/2DPAGE.html>
 8. Berlin Human Myocardial 2D Electrophoresis Protein Database,
URL: <http://www.chemie.fu-berlin.de/user/pleiss/dhzb.html>
 9. Human Colon Carcinoma Protein Database (Joint Protein Structure Lab) ,
URL: <http://www.ludwig.edu.au/www/jpsl/jpslhome.html>
 10. Yeast 2D-PAGE,
URL: <http://yeast-2dpag.gmm.gu.se/>
 11. Large Scale Biology Corp (2D maps: rat, mouse and human liver),
URL: <http://www.lsb.com/patterns.htm>
 12. PROTEOME Inc (YPD - Yeast Protein Database),
URL: <http://www.proteome.com/YPDhome.html>
 13. Keratinocyte database (Danish Centre for Human Genome Research),
URL: <http://biobase.dk/cgi-bin/celis/>
 14. NIMH-NCI Protein Disease Database (PDD) (plasma, CSF, urine),
URL: <http://www-pdd.ncifcrf.gov/>
 15. MitoDat - Mendelian Inheritance and the Mitochondrion,
URL: <http://www-lecb.ncifcrf.gov/mitoDat/>

The basic concept of flickering as a dynamic visualization technique is simple. If two images are perfectly alignable then one could simply align them. However many images such as 2D PAGE gels are non-uniform with 'rubber-sheet' distortion (i.e. local rotation, translation and magnification) -- with more distortion in some parts of the image than in other parts. These types of images may often be locally aligned by matching the morphology of local regions.

Even when they are aligned, it is still sometimes difficult to compare images which are quite different. Enhancing the images using various transforms before doing the flickering may help. Some of these transforms involve spatial warping, which maps a local region of one image into the geometry of the local region of another image while preserving its grayscale values. Other techniques include image sharpening and contrast enhancement. Image sharpening can be performed using edge enhancement techniques such as adding a percentage of the gradient or Laplacian to the original grayscale image. This works because the gradient and Laplacian have higher values at the edges of objects.

It is well known that 2D gels often suffer from local geometric distortions making perfect overlay impossible. Therefore, making the images locally morphologically similar while preserving their grayscale data may make them easier to compare. Spatial warping does not change the grayscale values of the synthesized warped image. Rather it samples pixels from the first input image and places them in the output image according to the geometry of a second input image. Changing the grayscale values would be counter to our goal of comparing image local structural differences since some structures (i.e. spots) might disappear and new artifacts might appear. The latter would be the case with using "morphing transforms" of two images which generates intermediate images where grayscale values are interpolated as well as the geometry.

Another useful operation is contrast enhancement which helps when comparing very light or very dark regions by adjusting the dynamic range of data to the dynamic range of the computer display.

2. MATERIALS AND METHODS

We describe a distributed gel comparison program called Flicker that is written in Java. Java is an general purpose, object-oriented programming language developed by Sun Microsystems [27]. It is similar to C++ and is rapidly becoming a standard for portable applications using the World Wide Web. The web sites <http://java.sun.com/> and <http://www.javasoft.com/> are the primary web sites for information on Java. A list of official books on Java published by SUN is on <http://java.sun.com/books/Series>. Java has been optimized for use as a web programming language and is used to write both web-based ``applets" and stand-alone "application" programs. A Java applet is the term

Sun coined for a mini-application that runs inside of a browser when it loads the web page containing the URL to the applet. A Java application on the other hand runs by itself on a user's computer. When invoked in a user's Java-capable web browser (such as Netscape) by clicking on the URL for an applet, the applet is automatically downloaded from a WWW server and run in the browser.

Java now gives us the ability to do real-time comparisons of local 2D gel image data on the user's computer with gel images residing in various remote databases on the Internet.

The Flicker program may be used as either a Java applet or a stand-alone Java application. Java applets may be viewed on any Java-capable web browser. Alternatively, Java applications may be run on various machines using the Java applet-viewer available free from SUN Microsystems for most systems including UNIX, Windows PCs and Macintosh computers. [Figure 1a](#) illustrates the static web paradigm commonly used by all web servers prior to the introduction of Java. It shows the concept of the Internet client-server with the user's web browser as the client and the web server as the Internet server. The server performs some service (such as returning data) in response to a request from the client. Figure 1b illustrates the second generation extended web paradigm which distributes the computing between the user's client machine (housing the Java-capable web browser) and the remote web-database servers. Here, computation and interaction can take place on the client's computer - independent of the server. [Figure 1c](#) illustrates the relationship between the user's image data, the Flicker program, and image data from the remote 2D gel database.

The user interacts with the Flicker web server using their web browser using the client/server paradigm. The web browser may be thought of as a client which makes requests of a 2D gel database web server. When Flicker is invoked by the user in their Web browser, it downloads the Flicker JAVA program from our web server into the user's web browser. Then, regardless of whether it gets data from that web server, other web servers or locally, it will load two images.

a) Static client-server paradigm

user's Web browser <----- Internet -----> Web server

- Steps: 1. Request for data ==>
2. Data returned from server <==

b) Dynamic client-server paradigm

user's Web browser ><----- Internet -----> Web server

- | | | |
|--|-----|---------------------------|
| Steps: 1. Request for Java Applet | ==> | |
| 2. Java Applet returned on Web browser | <== | 3. Java Applet started |
| 4. Applet requests data | ==> | |
| 5. Data returned from server | <== | 6. Data processed locally |

c) Distributed data client-server paradigm

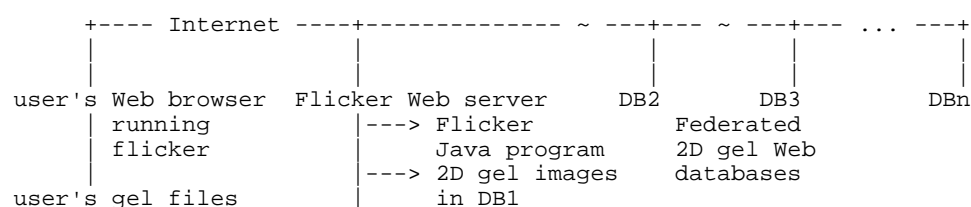


Figure 1. Static and dynamic client-server web paradigms. **a)** Illustrates the static web paradigm commonly used by all web servers prior to the introduction of Java. Although not shown, the Common Gateway Interface (CGI) of the HTTP web protocol which supports interactive forms and image maps still effectively results in static data and so is included in the static paradigm. **b)** Illustrates the extended web paradigm which distributes the computing between the user's client machine (housing the Java-capable web browser) and the web database server machine. In this model, computation and tightly-coupled user interaction can take place on the client's computer - *independent* of the server. **c)** Illustrates the relationship between the user's web browser with local 2D gel images, the web server which contains the Flicker program, and the federated 2D gel databases (DB) on other web servers. Two gels to be compared may come from the Internet Web databases or from the user's local file system. The images may be from either the Flicker 2D gel image DB web server or from other federated 2D gel image web databases DB2, DB3, ...DBn. For example, DB2 might be the SWISS-2DPAGE, DB3 might be the CSH Labs Quest Yeast protein database, etc.

We will now describe the capabilities of the Flicker applet including some discussion of the image processing algorithms. We then illustrate the use of Flicker with an example of the comparison of corresponding subregions of two images of human plasma 2D gels run in different laboratories under different conditions.

2.1 Use of Flicker for comparing images

When flickering two images with the computer, one aligns putative corresponding subregions of the two rapidly alternating images. The flicker display overlays the same space on the screen and are aligned by interactively moving one image relative to the other. Using the mouse, the user initially selects what they suspect is the same prominent spot or object in similar morphologic regions in the two gel images. The images are then centered in the flicker window at these spots. When these two local regions come into alignment, they appear to "pulse" and the images fuse together. At this point, differences are more apparent and it is fairly easy to see which spots or objects correspond, which are different, and how they differ. We have found that the user should be positioned fairly close to the flicker window on the screen to optimize this image-fusion effect.

The proper flicker delays, or time each image is displayed on the screen, is critical for the optimal visual integration of image differences. We have also found that optimal flicker rates are dependent on a wide variety of factors including: amount of distortion,

similarity of corresponding subregions, complexity and contrast of each image, individual viewer differences, phosphor decay-time of the display, ambient light, distance from the display, etc. In addition, the process of flickering images is easier for some people than for others. When comparing a light spot in one gel with the putative paired darker spot in the other gel one may want to linger longer on the lighter spot to make a more positive identification. Because of this, we give the user the ability to set the display times independently for the two images (typically in the range of 0.01 second to 1.0 second with a default of 0.20 second) using separate delay scroll bars located under each image. If the regions are complex and have a lot of variation, longer display times may be useful for both images. Differential flicker delays with one longer than the other are also useful for comparing light and dark sample gels. Changing image brightness and contrast also is useful when flickering and the Flicker program has provision for interactively changing these parameters as well.

[Figure 2](#) shows the screen of the Flicker applet. Two images are loaded into the lower scrollable windows. Only part of the image is visible in a scrollable window and this subregion is determined by horizontal and vertical scroll bars. This lets the user view any subregion of the image at high resolution. These images may be navigated using either the scroll bars or by moving the mouse with the button pressed in the scrollable image window. Then, each image in the flicker window is centered at the point last indicated in the corresponding scrollable image window. A flicker window is activated in the middle of the screen when the `Flicker` check box is set. Images from the left and right scrollable images are alternatively displayed in the flicker window.

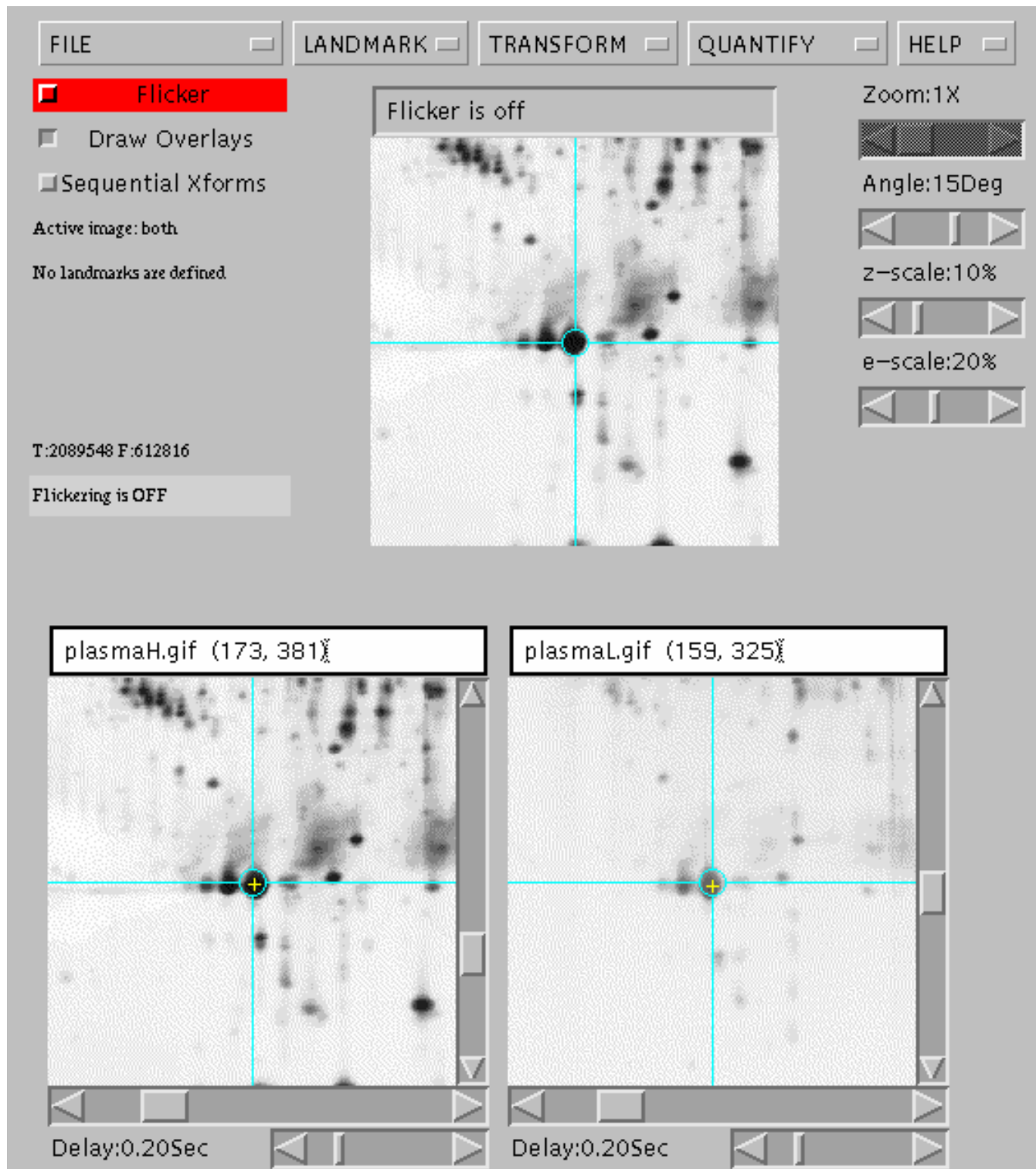


Figure 2. Screen view of Flicker Java applet run in the Netscape browser. Control menus at the top invoke file operations, landmarking, spatial warping, grayscale transforms and object quantification. Scroll bars on the side determine various parameters used in the transforms. Additional options allow resetting images, and loading new images from files on the Internet or the local file system, aborting the current transform in progress, and help. Check boxes on the left activate flickering and specify display options. The flicker image is in the middle of the frame when it is enabled. The two labeled human blood plasma gel images are shown in the bottom

windows which have scroll bars to position the region of interest and flicker time-delays which are used when flickering. Image plasmaH is an IPG gel from Denis Hochstrasser's Lab in Geneva and plasmaL is a non-IPG gel from Carl Merril's Lab at NIMH. When images are transformed, the results are shown in the same scrollable windows.

2.2 Description of the image processing methods

As mentioned, there are a number of different image transforms which can be invoked from the control panel. These are useful for changing the geometry, contrast or object definition to make it easier to compare potentially corresponding regions. As we go through the transforms we will indicate how they may be used. Some affect one image while some affect both. Flickering may be activated at any step in the image processing to compare the current transformed images.

The spatial warping transforms require defining several corresponding landmarks in both gels. As we mentioned, one gel image can be morphologically transformed to the geometry of the other using the affine or other spatial warping transformations. They map the selected image to the geometry of the other image. A "trial" landmark is defined by clicking on an object's center anywhere in a scrollable image window. This landmark would generally be placed on a spot. [Clicking on a spot while pressing the `CONTROL` key realigns that image in the scrollable window so the selected spot is in the center.] After defining the trial landmark in *both* the left and right windows, selecting the `Add Landmark` option in the `{Landmark` menu defines them as the next landmark pair and identifies them with a red letter label in the two scrollable image windows. Selecting the `Delete Landmark` option deletes the last landmark pair defined.

The `TRANSFORM` menu has a number of selections which include warping, grayscale transforms and contrast functions. There are two warp method selections: `Affine Warp` and `Poly Warp` which are performed on only one image (the last selected by clicking on an image). Unlike the warp transforms, the grayscale transforms are performed on both images. These include: `Pseudo 3D`, `SharpenGradient`, `SharpenLaplacian`, `Gradient`, `Laplacian`, `Average`. The contrast functions are `Complement`, and `ContrastEnhance`. These functions are described below in more detail.

Finally, there are several object quantification options. These allow the user to draw a boundary around an object and then compute simple features such as centroid, total area and grayscale, etc. The `QUANTIFY` menu includes: `Measure Background`, `Measure Object` and `Measure Done`.

2.2.1 Affine transform spatial warping

The affine transform is used to spatially warp the last selected image to the geometry of

the other image. This might be useful for comparing gels which have some distortion, were of different sizes or were run under slightly different conditions. It is used as an *inverse* mapping as described in [28]. Let $(u,v)=f(x,y)$, where (x,y) are in the output image, and (u,v) are in the input image. Then, in a raster sweep (x,y) through the output image, pixels are copied from (u,v) in the input image to the output image. The affine transformation is given in equations (1) and (2):

$$(1) \quad u_{xy} = ax + by + c$$

$$(2) \quad v_{xy} = dx + ey + f$$

The system of 6 linear equations is solved for coefficients (a,b,c,d,e,f) using three corresponding landmarks in each gel. So three landmarks must be defined before doing this transform. The program checks to ensure that the landmarks are not co-linear and does not do the transform if they are. Co-linear landmarks cause the set of equations to be ill-defined and prevent a solution from being found.

2.2.2 Polygonal transform spatial warping

Similarly, the polygonal transform is used to spatially warp the last selected image to the geometry of the other image. It is used as an *inverse* mapping as described in [28]. Again, let $(u,v)=f(x,y)$, where (x,y) are in the output image and (u,v) are in the input image and are defined in equations (3) and (4). Then in a raster sweep through the output image, pixels are copied to (x,y) in the output image from (u,v) in the input image.

$$(3) \quad u_{xy} = \text{Sum}_{i=0:n} (\text{Sum}_{j=0:n-1} a_{ij} x_i y_j)$$

$$(4) \quad v_{xy} = \text{Sum}_{i=0:n} (\text{Sum}_{j=0:n-1} b_{ij} x_i y_j)$$

For these equations, we use n equal to 2 resulting in a set of non-linear functions. [The affine transform is a linear polygonal transform with a value n equal to 1.] Then, the system of 12 equations is solved using a weighted least square error method described in [28]. Therefore six corresponding landmarks in each gel are required to solve the equations. So six landmarks must be defined before doing this transform. Because of the more accurate warping model, this transform should do a better job of warping, but at a cost of having to define more landmarks and more computation.

2.2.3 Pseudo 3D transform

The Pseudo 3D transform is a *forward* mapping that generates a pseudo 3D relief image to enhance overlapping spots with smaller spots seen as side peaks. The gel size is *width* by *height* pixels. The gray value determines the amount of y shift scaled by a percentage

(scroll bar z_{Scale} (in the range of 0 to 50%). Pseudo perspective is created by shifting the image to the right (left) by scroll bar angle $theta$ (in the range of ± 45 degrees). The transform is given in equations (5) through (8).

$$(5) \theta_{rad} = (180/PI) * \max(-45, \min(\theta_{deg}, 45))$$

$$(6) d_x = width * \sin(\theta_{rad}),$$

then

$$(7) x' = (d_x * (height - y)/height) + x,$$

$$(8) y' = (y - z_{Scale} * g(x, y)),$$

where $g(x, y)$ is in the the original input image and (x', y') is the corresponding position in the output mapped image. Pixels outside of the image are clipped to white. The pseudo 3D transform is applied to both images so that one can flicker the transformed image.

2.2.4 Image contrast enhancement

Sometimes gels are too dark or too light. Grayscale image contrast may be enhanced using either manual interaction or histogram equalization. Manually, the user changes the contrast (horizontal) and brightness (vertical) for both images simultaneously by dragging the mouse while pressing the `SHIFT` key. Histogram equalization first computes the histogram of gray values in the images and then maximizes the pixel values of the full range of possible gray values. If g_{max} and g_{min} are the extrema found in the dynamic grayscale range of an image and $[g_{white}:g_{black}]$ is the maximum possible range, then the image may be rescaled by equation (9):

$$(9) g' = ((g - g_{min}) / (g_{max} - g_{min})) * (g_{black} - g_{white}).$$

2.2.5 Edge sharpening

Edge sharpening may be useful for sharpening the edges of fuzzy spots. It is done by adding a percentage of a 2-dimensional edge function of the image to original image data $g(x, y)$ as shown in equation (10). The edge function increases at edges of objects in the original image. Typical edge functions include the gradient and Laplacian. The scroll bar e_{Scale} value (in the range of 0 to 50%) is used to scale the amount of edge detection value added.

$$(10) g'(x, y) = (e_{Scale} * edge(x, y) + 100 - e_{Scale}) * g(x, y) / 100.$$

For example, the gradient is approximated as the maximum of four 8-neighbor difference function convolution filters f_{angle} denoted as $f_0, f_{45}, f_{90}, f_{135}$ for 3x3 pixel regions.

$$\begin{array}{lcl}
 f_0 = \begin{array}{ccc} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{array}, & f_{90} = \begin{array}{ccc} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{array}, \\
 f_{45} = \begin{array}{ccc} 0 & +1 & +2 \\ -1 & 0 & +1 \\ -2 & -1 & 0 \end{array}, & f_{135} = \begin{array}{ccc} +2 & +1 & 0 \\ +1 & 0 & -1 \\ 0 & -1 & -2 \end{array}
 \end{array}$$

The four convolution filters f_k , for angle k , show increased values for changes in four compass directions 0, 45, 90, 135, if the orientation of an edge matches one of the filters. They are applied over the local pixel neighborhood defined by (i,j) in the range of $[-1:+1, -1:+1]$. Then, they are applied globally. For each pixel (x,y) in the image and for each filter f_k , the computation of $grad_{xy}$ is given in equations (11) and (12) with d_{kxy} being the difference value associated with angle k ,

$$(11) \quad d_{kxy} = \text{Sum}_{i=-1:1} \text{Sum}_{j=-1:1} f_{kij} g_{x+i, y+j}$$

$$(12) \quad grad_{xy} = \max(|d_0|, |d_{45}|, |d_{90}|, |d_{135}|)$$

Similarly, the Laplacian convolution filter f_{ij} is approximated as the absolute value of the difference between the central pixel and the sum of its 8 neighbors. The Laplacian convolution is computed over the entire image similar to the gradient computation. The Laplacian filter f_{ij} function is:

$$f(i, j) = \begin{array}{ccc} -1 & -1 & -1 \\ -1 & -8 & -1 \\ -1 & -1 & -1 \end{array} / 9$$

3. RESULTS

We now illustrate Flicker with several computer screens taken from an interactive session of two plasma protein 2D PAGE gels. Image plasmaH is an IPG gel from Denis Hochstrasser's Lab in Geneva and plasmaL is a non-IPG gel from Carl Merril's Lab at NIMH. The plasmaL is locally warped to the geometry of plasmaH using the affine transform. The landmarks required for the transform are shown in [Figure 3a](#) and the resulting affine transformed image in [Figure 3b](#). Note the subtle shift in position of the landmarked spots in the plasmaL gel. If we were flickering these two subregions they would align better than without the transform. [Figure 4a](#) shows the results of doing a pseudo 3-dimensional transform of the two images. [Figure 4b](#) shows the results of using the SharpenLaplacian transform which improves the definition of the edges on some of

the lighter fuzzy spots.

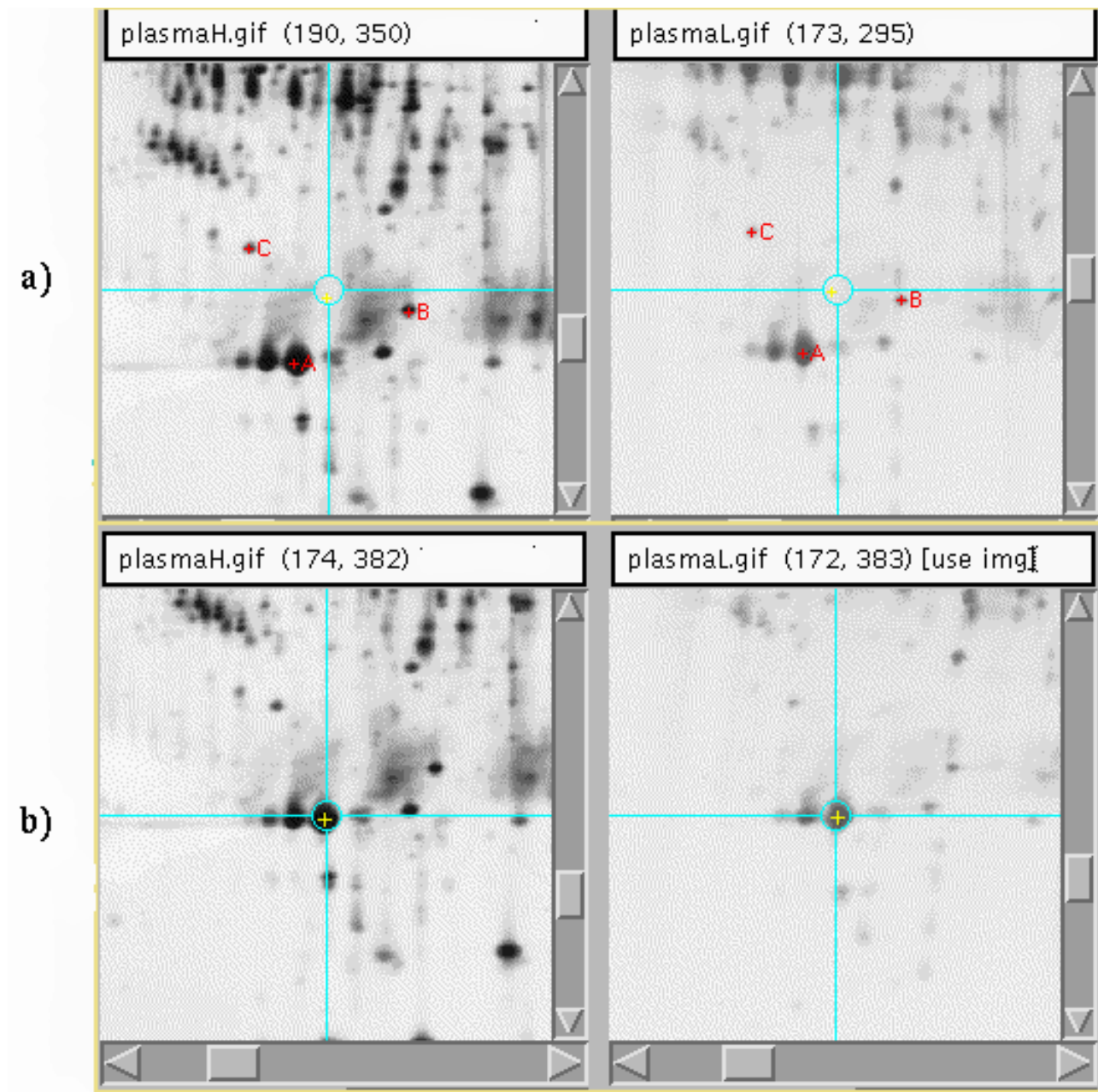


Figure 3. Affine transform of human plasma gel image. The transform warps the geometry of a local region defined by the three landmarks so it more closely resembles the geometry of the corresponding local region in the other gel. **a)** Screen view of scrollable windows from Flickr with 3 landmarks defined in both gel images. This was done in preparation for doing the affine image transform. The corresponding landmark spots were selected so that they were unambiguously defined in both gel images. **b)** images after the affine warp transform of the right (plasmaL, non-IPG) image to the geometry of the left (plasmaH, IPG gel) image. The 3 landmarks in panel (a) were used to solve equations (1) and (2) for (a, b, c, d, e, f) . The output image is then visited at each pixel (x, y) and the transform used as an inverse mapping to get pixel values $g(u, v)$ to save in the output image $g(x, y)$.

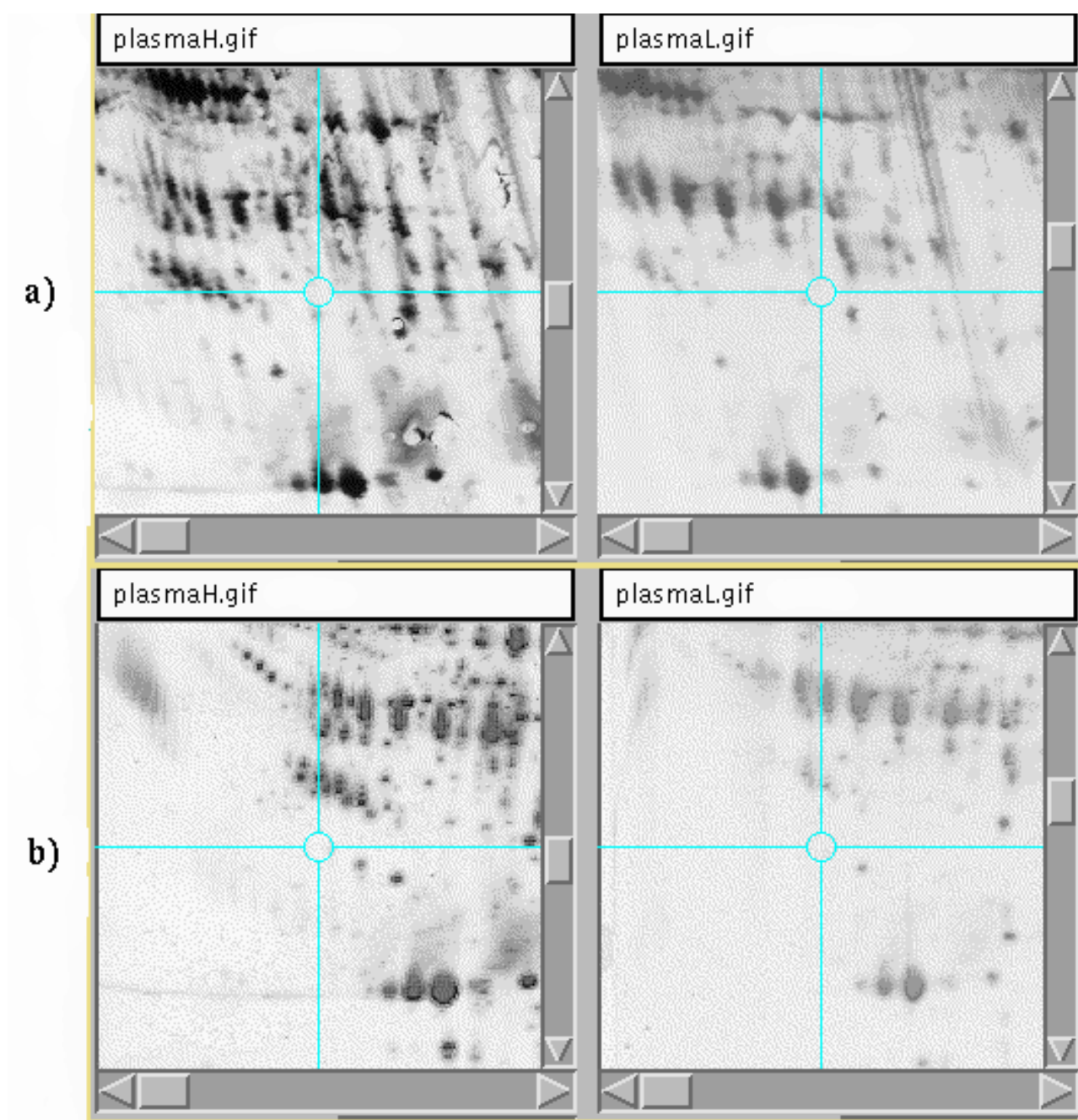


Figure 4. Grayscale image transforms applied to both human plasma gel images. **a)** Screen view of scrollable image windows from Flicker after the pseudo 3D transform applied to both gel images at the default settings for *angle* and *zScale* of 15° and 10% respectively. **b)** images after the SharpenLaplacian transform applied to both gel images at the default setting for *eScale* of 20%.

4. DISCUSSION

Flicker is a WWW image comparison Java program running on the user's local computer. It accepts images from both federated web image databases as well as images

from the user's computer. Using such distributed image databases now offer the possibility of comparing biomedical data from user's and federated Internet databases. Also, Java is a way to easily distribute substantial real-time software running on user's local computers to those computers. Although there are problems with the current Java paradigm, these are being reduced or eliminated. This new data analysis paradigm now allows scientists to more easily collaborate and compare data over the World Wide Web.

The advantages of this technique are it embodies a low cost existing technology which requires little user effort and it saves time over the alternative ways of comparing 2D gels. Increasingly, web access is available to a wide variety of federated biomedical databases (cf. [Table 1](#)). Recent advances in technology make scanning images of user's 2D gel images with an inexpensive scanner more accessible and relatively easy.

However, there are also disadvantages in comparing gels this way. It is only good for doing a rough comparison and there is currently no simple way available to do quantitative comparison -- although we are working on the latter. The latter would be possible with any of the 2D gel computer database systems referenced in the Introduction, but with additional expense and effort. One should keep these limitations in mind when using the technique.

The intent of applying image transforms is to make it easier to compare regions having similar local morphologies but with some different objects within these regions. Image warping prior to flickering is intended to spatially warp and rescale one image to the "shape" of the other image so that we can compare them at the same scale. This should help make flickering of some local regions on quite different gels somewhat easier. There are two warping transforms, affine and polynomial, requiring 3 and 6 landmarks respectively. For those cases where the gels are fairly similar, the user may be able to get away with using the simpler (affine) transform. Other transforms including image sharpening may be useful in cases where spots are very fuzzy, as in the case of Southern blots. When two corresponding local regions of the two images are radically different (eg. when gels are run differently as: IPG vs. non-IPG, gradient vs. non-gradient SDS), then even using these transforms may not help that much.

In cases where there is a major difference in the darkness or lightness of gels, or where one gel has a dark spot and the other a very faint corresponding spot, it may be difficult to visualize the light spot. By differentially setting the flicker display-time delays, the user can concentrate on the light spot using the brief flash of the dark spot to indicate where they should look for the light spot. We have found differential-flicker to be very helpful for deciding difficult cases.

The use of zoom also may be helpful if there are a number of small spots in the regions

being compared and they are difficult to distinguish with the default 1X magnification.

There are several problems with Java as it is implemented with current web browsers because of restrictions due to applet security concerns. Because of fears of security breaches, Netscape and other web browser providers have disabled Java applets running on their browsers from reading or writing local files. They also restrict access of web URLs to the host computer where the Java program originated (i.e. in this case, the site where the Flicker program itself comes from). Unfortunately, this prevents the Flicker applet from loading your local image files or other federated databases not on the Flicker web server. It thus prevents you from comparing data from different sources. However, there are two ways to get around the security problem. The stand-alone Java applet viewer and running Flicker as a stand-alone java application rather than as an applet do not have these security restrictions. The applet-viewer and java application interpreter are available as part of the free Java Development Kit (from SUN Microsystems for UNIX, Windows PCs and Macs). In addition, major computer vendors will be bundling Java into their standard operating systems in the future. Therefore, it is currently possible to use Flicker to compare gels from different sources while we wait for the current security restrictions are removed from future web browsers. In the long term, browser vendors are developing methods using "access control lists" to both protect your local systems and give users access to local and federated data.

There are several other restrictions. The current Java library only handles GIF and JPEG image formats. Images in other formats such as TIFF need to be converted to GIF format. This will be remedied in the future. Because we are doing image pixel processing with the Flicker program, it requires more memory and computations than applications that only manipulate text. Therefore it may require a more powerful CPU and more memory than some users currently have.

Although Flicker is not currently able to quantitate spots, it may be possible to do so in the future using this generic gel image data. The problem is primarily in the calibration of grayscale to optical density (OD) or counts per minute (CPM) in order to correctly compute integrated density. This assumes that the optical density of a gel image is proportional to protein concentration. It would be helpful to have federated web 2D gel databases publish the OD (grayscale to OD) and CPM calibrations along with each image. Image spot segmentation and quantitation require a lot of memory, so there may also be possible problems with memory limits for some user's computers.

Because the technique compares images using a domain-independent visual comparison, there are a number of other problem domains where this technique might be applied. These include: gel electrophoresis (both 1-D and 2-D for protein, RNA and DNA materials), chromatographs, serial section images produced by various microtome

methods, medical X-rays for comparing bone growth or tumor progression, MRI or PET images, astronomical star maps, spectra (of anything), graphs (of anything), problem domains which produce mis-aligned or distorted images, problem domains which lose alignment during data acquisition.

Of the features we have mentioned, some are not fully functional and we are working to resolve this. These problems include: file and general URL access, zoom, interactive contrast enhancement, data measurement and calibration functions, memory overflow, invoking Flicker as a Java application. The current version uses more memory than it might and we are working on recoding the program to make more optimal use of memory resources. This is important, since many users may not have a lot of memory on their computers to run this program or all of its features.

We are exploring future improvements to Flicker to include: collaborative viewing similar to our Xconf image conferencing work. It may be possible to integrate the Flicker program with the web groupware Habanero Project [The Habanero Project at NCSA is investigating collaborative work environments by recasting single user computer software tools as multi-user tools. URL: <http://www.ncsa.uiuc.edu/SDG/Software/Habanero>] so that several remote users could share a Flicker session; investigating the same data spot quantitation, MW/pIE calibration for 1D and 2D gels; automatic alignment of spot clusters (local morphologic regions) between gels; and access to and integration with 2D gel quantitative, qualitative, and disease databases using federated web databases. The Flicker applet is available on our WWW server at the following URL <http://www-lecb.ncifcrf.gov/flicker/>.

Thanks are due to Tom Schneider, Ellen Burchill, and Greg Thornwall for useful suggestions for improving the GUI and this manuscript.

REFERENCES

1. Berners-Lee, T.J., Cailliau, R., Luotonen, A., Henrick. F., Secret, A., *Comm. Assoc. Comp. Mach.* 1994, **37**, 76-82.
2. Appel, R.D., Sanchez, J.-C., Bairoch, A., Golaz, O., Miu, M., Vargas, J.R., Hochstrasser, D.F., *Electrophoresis* 1993, **14**, 1232-1238.
3. Appel, R.D., Sanchez, J.-C., Bairoch, A., Golaz, O., Rivier, F., Pasquali, C., Hughes, G.J., Hochstrasser, D. F., *Nucleic Acids Res.* 1994, **22**, 3581-3582.
4. Sanchez, J.-C., Appel, R.D., Golaz, O., Pasquali, C., Rivier, F., Bairoch, A., Hochstrasser, D.F., *Electrophoresis* 1995, **16**, 1131-1151.
5. Jungblut, P., Thiede, B., Zinny-Arundl, U., Muller, E-C., Scheler, C., Whittmann-Liebold, B., Otto, A., *Electrophoresis* 1996, **17**, 839-846.
6. Wilkins, M.R., Sanchez, J-C., Williams, K.L., Hochstrasser, D.F., *Electrophoresis* 1996, **17**, 830-838.

7. Celis J. E. (Ed), Special issue: Two-dimensional Electrophoresis protein databases, *Electrophoresis* 1992, **13**, 891-1062.
8. Celis J. E. (Ed), Special issue: Two-dimensional Electrophoresis protein databases, *Electrophoresis* 1993, **14**, 1089-1240.
9. Celis J. E. (Ed), Special issue: Electrophoresis in Cancer Research, *Electrophoresis* 1994, **15**, 305-556.
10. Celis J. E. (Ed), Special issue: Two-dimensional Electrophoresis protein databases, *Electrophoresis* 1995, **16**, 2175-2264.
11. Bjellqvist, B., Hughes, G.J., Pasquali, C., Paquet, N., Ravier, F., Sanchez, J.-C., Frutiger, S., Hochstrasser, D.F., *Electrophoresis* 1993, **14**, 1023-1031.
12. Pallini, V., Bini, L., Hochstrasser, D., Proceedings: 2D electrophoresis: from protein maps to genomes. Univ. of Siena, Italy, Sept 5-7, 1994.
13. Taylor, J., Anderson, N.L., Scandora, A.E., Willard, K.E., Anderson, N., *Clin. Chem.* 1982, **28** 861-866.
14. Lipkin, L.E., Lemkin, P.F., *Clin. Chem.* 1980, **26**, 1403-1413.
15. Lemkin, P.F., Lester, E.P., *Electrophoresis* 1989, **10**, 122-140.
16. Appel, R.D., Hochstrasser, D.F., Funk, M., Vargas, J.R., Pellegrini, C., Muller, A.F., Scherrer, J.-R., *Electrophoresis* 1991, **12**, 722-735.
17. Olson, A.D., Miller, M.J., *Anal. Biochem.* 1988, **169**, 49-70.
18. Garrels, J.I., Farrar, J.T., Burwell IV, C.B., in Two-Dimensional Gel Electrophoresis of Proteins, J.E. Celis, R.Bravo (Eds), Academic Press, NY, 1984, pp 37-91.
19. Vincens, P., Rabilloud, T., in Galteau, M.M., Siest, G. (Eds), Recent Progress in Two-Dimensional Electrophoresis, University Press of Nancy, France, 1986, pp 121-130.
20. Lemkin, P., Merrill, C., Lipkin, L., Van Keuren, M., Oertel, W., Shapiro, B., Wade, M., Schultz, M., Smith, E., *Comput. Biomed. Res.* 1979, **12**, 517-544.
21. Lemkin, P.F., *Comput. Biomed. Res.* 1993, **26**, 1-27.
22. Lemkin, P.F., Orr, G., Goldstein, M.P., Creed, J., Whitley, E., Myrick, J.E., Merrill, C.R., *Electrophoresis* 1995, **16**, 1175.
23. Merrill, C., Goldstein, M., Myrick, J., Creed, J. and Lemkin, P.F., *Appl. Theor. Electrophor.* 1995, **5**, 49-54.
24. Lemkin, P., Orr, G.A., Goldstein, M., Creed, J., Myrick, J., Merrill, C., *Appl. Theor. Electrophor.* 1995, **5**, 55-72.
25. Lemkin, P., Chipperfield, M., Merrill, C. and Zullo, S., *Electrophoresis* 1996, **17**, 556-572.
26. Appel, R.D., Bairoch, A., Sanchez, J.-C., Vargas, J.R., Golaz, O., Pasquali, C., Hochstrasser, D.F., *Electrophoresis* 1996, **17**, 540-546.
27. Arnold, K., Gosling, J., The Java Programming Language. Addison Wesley, New York, 1996.
28. Wolberg, G., Digital Image Warping, IEEE Computer Press Monograph, Los

Alamitos, CA, 1990.