Multi-Isotope Imaging Mass Spectrometry Workshop
National Resource for Imaging Mass Spectrometry
Cambridge, MA

September 24, 2003
Speaker: Douglas Benson

Note: Transcript has been edited for clarity, but has not yet been reviewed by the author. Accordingly, inaccuracies may be present. The author-edited version of the transcript will be posted at a later date.

© 2004 by the author. This information may not be used without referencing the author.

Douglas Benson: First, I'd like to thank Claude for inviting me today. And I want to keep this as conversational as possible. So at any point any time anyone has any questions please stop, interrupt, and let's make sure we get any questions answered. I'm going to talk about – well, actually, I'm going to talk mostly about the ratioing method that Claude presented yesterday.

But I also wanted to give a bit of a brief demonstration of Image J, which is an application developed at NIH, that we've developed some software to use the MIMS images, the Cameca format images with. And if time permits we'll go in a little bit on another software package, IC, which was developed by a company, (Inivision), that I worked for some years ago.

I started developing software for image analysis, or for imaging applications as a graduate student about 25 years ago. I was working in a lab that had a fluorescent dye that would measure oxygen quantitatively. And we came up with this wild idea that you could put a television camera on a microscope, and quantitate fluorescence under the microscope.

A typical experiment for us was collecting data literally on videotape, carting a videotape player across campus to a computing center. Digitizing the video a frame at a time, and transferring it to an IBM 360 to do the data processing. So real time for us was on the order of three to five days. And the total cost of the system was probably a couple million dollars, in those 1975 dollars. So things have changed quite a bit in the last couple of years.

About five years after that, I was at a conference in Pittsburgh at the Carnegie Mellon Institute. And there were about five labs meeting at a workshop very similar to this discussing ways of quantifying or quantitating fluorescence in a microscope. And I wanted to bring that up because this group reminds me very much of the same situation 15, 20 years ago. I think where fluorescence microscopy, and quantitative fluorescence microscopy was then is where you are now. And I think one of the points George made the day before yesterday, about that there are a lot of techniques that have been applied to looking at problems in fluorescence microscopy that are applicable to ion imaging problems, and the kinds of problems you are facing.

Very briefly, I wanted to give to everyone – some of you, or many of you may already be familiar with these books, but there are several books out there that I think are worth looking into for those who have little or minimal experience in image processing, and imaging software (Slide #2). The first one, Castleman, is probably the gold standard that's been around for about 20, 25 years. There's a second rev of it out that came in 1996. It's a little more engineering, designed for engineers, so it's probably a little technical for the beginner.

The second book, John Russ's *Image Processing Handbook* – actually, John Russ has a material science background, so he's probably – this may be a little better suited for a crowd like this. But it doesn't go into some of the technical or theoretical detail that Castleman and Pratt's books do.

Why ratio imaging? (Slide #3) As I mentioned some years, 15 years ago, when we had this conference at Carnegie about fluorescence imaging and fluorescence applications, there was one technique that came out during that conference that has survived the test of time, and that

was ratio imaging. And there are lots of ways to try to quantify data in any kind of optical or ion microscope system, but ratioing turns out to be one that, again, is the most reliable, or in my opinion, my experience has been most reliable.

The advantages of, it allows you to normalize for spacially variant, systematic errors. And the results can be expressed relative to internal controls. In your case, because you know the ratio of (terrestrial) isotopes to high precision, you have an added advantage of using this method.

The disadvantage is that you get extremely high dynamic range images (Slide #3). Typical display technologies can give you 256 shades of gray. The eye can see maybe 100. And there are some people that would argue more, but I think the evidence is that you can probably see 60, maybe 100 shades of gray. You can see thousands of colors, but to be able to distinguish details just looking at a gray scale image, it's very difficult.

The second problem or disadvantage with ratioing is that you have a high noise component for areas where there are statistically insignificant numbers of counts.

I wanted to go over a model that we use for fluorescence applications, and I asked George to think about this before I started. So this is not a problem for you, but at least it gives you a basis, or something to think about when you start applying some of these techniques (Slide #4).

The fluorescence you observe at any point – x,y location – in the screen, is a function of the concentration of a fluorescent dye. It's a function of the excitation intensity. It's a function of the quantum yield of the (fluorofore), the extension coefficient at the wavelength that you're exciting, and some kind of global gain factor which, again, is spatially variant. And it's complicated by the fact that in most microscope systems, or (light) microscopy applications, you have the background component which is not – it's an additive component. It's not simply a multiplicative component.

So the computer ratio with a system like that, you have to first subtract that background from the two signals – that were, in this case, species one, and species two; in your case mass one, or mass two – and to remove the additive component before you calculate the ratio.

The important piece of information here is that spatially variant terms, like the (beam) current, or any gain components in the system cancel out during the ratio. Is everyone clear about that?

This is an example of a mass 27 (Slide #5). This is the (C) nitrogen 15. Is this a (mouse embryo)?

XX: (Fibroblast).

DB: (Fibroblast). Thank you. And you can see in this series of images the lower image, the denominator, has a very high count, the numerator has a very low count, a mean of 2.9. And the corresponding ratio has a very noisy component to it. Right?

Now just to demonstrate I think what Frank was showing yesterday using pseudo-color, or some color scale mechanism, you can – in some cases you can see more detail because of our ability to perceive more different colors versus grayscale. But it actually (confuses) the image frequently because the noise – you have areas of high ratios which are from statistically insignificant regions (Slide #6). Right?

I wanted to zoom in just to show a case of where you have there's clearly some detail. This was the denominator, which has high counts. And just to show an area that's been zoomed up, and the corresponding numerator/denominator and ratio from that (Slide #7). And you can see, for all practical purposes, there's no spatial information in that ratio image (Slide #8).

So what we want to do is we want to somehow preserve the pixels that have significant numbers of counts, and minimize or reduce the information from the pixels that have very few counts. And this is where this method of the HSI (transform) comes in. And that's an example just to give everyone a bit of a hint as to where we're going (Slide #9). So what is HSI?

Claude Lechene: Go back, please? It's a little fast. Go to the previous one?

XX: It was just too fast for me to –

George Slodzian: (We do not have) ??. We see (the picture) ??.

CL: No, no. The clue is to compare with the previous one where you didn't have any details (on the right). This one. So the one you will see now is equivalent to the one you saw before.

DB: Yes.

CL: They're the same.

XX: ?? don't understand why ?? (part of the image is darker).

CL: This is what he will explain now, I hope.

DB: It's software magic Okay. Hue, saturation, and intensity is actually – it's a transform that's been around since television, I guess. I think it's the knobs – for those of us who remember the early color TVs, you had these knobs; the tint, color, and brightness knobs. It's basically the same thing. An analog scale for doing the same thing.

Hue – the critical point I want to make here is that hue is actually a radial measurement, so it has units of degrees. And if I extended the scale – which I (did) on the right, the color bar – actually as you go to 360 it becomes the same color as it is in zero (Slide #10). So it's a circular component to it.

Saturation, I'm not really interested in the details here, but I just wanted to – saturation is actually a measure of how much color there is. Probably the easiest way to think about this is red saturated turns pink. And the more saturation, it goes from pink to white. So we're not – in our case I'm not really interested in the saturation. What I am interested in is coming up with

some color scale, and some means of modulating the color, which is using the intensity term.

So how do we do this? The first thing that you have to do is you want to convert ratios, which are some floating point numbers. And in the case of the images coming from a (sume) system, the images are scaled between 0 and 65,000 gray levels. So we choose the same scale just to preserve the data precision of the original data. And so we come up with some indexed ratio, which is the calculated ratios normalized to some range of ratios, and scale that to this integer between 0 and 65,000 (Slide #11). Is that clear?

CL: No.

XX: What's the 65,000?

DB: Why the 65,000?

XX: ??.

DB: It's 16 bits.

XX: It just has to do with the file format.

DB: You have to come up with some mechanism of scaling the ratios to color. So it comes up with just a simple scale table for doing the conversion.

We also convert the hue, this range of colors, to a corresponding range of gray levels, or a range of numbers, indexes. And I wanted to show here that actually what computers understand are levels of red, green, and blue. So you convert a color scale, as we have on the left, to a series of red, green, and blue components (Slide #12). Is that clear?

And this is the magic (Slide #13). For each pixel you calculate the ratio. And the ratio – you calculate that indexed ratio that I showed two slides before. And from the indexed ratio you get a hue for the red, green, and blue for that pixel. So that gives me the color.

At the same time, you calculate the intensity, based on some function of the numerator and denominator. Typically we use the maximum. So depending on how many counts were in the numerator, or the number of counts in the denominator, you modulate the color based on that number of counts.

Anders Meibom: What I don't understand is you have one ratio, and all of the sudden you have three colors (assigned to that).

DB: Right. You have one color, it's just that colors have a red, green, and blue component.

William Lamberti: I'm trying to digest this real quickly. It sounds like you're going to basically an indexing scheme, where you're simply trying to assign a unique color to a unique gray level.

DB: A unique ratio.

WL: A numerical result.

DB: Right.

WL: So you have a set of ratios that are resulted from your calculation. Okay. With that data set of numbers, (which are) floating point, you're now trying to assign a unique –

DB: Color to each ??.

WL: Given the number of colors that you have, which in this case is 65,000 colors. How many colors are you using?

DB: 16 million.

WL: 16 million in this case. Okay. Each R, G, and B has 16 bits of variability?

DB: Yes. It actually turns out on the – it's scaled down to 8 bits when you go to a display system. But for internal calculations, for precision, we keep everything at the same precision as the original data.

WL: Okay, thanks.

DB: So on this scale on the bottom, what you see is the ratio, going from a minimum to a maximum, changes from blue to pink. The intensity changes from black to some continuous color (Slide #13). I see a number of heads nodding there.

GS: The ratio is ??.

DB: Right.

GS: So ??.

DB: So ratios vary from blue to pink.

GS: For instance ?? (purple) becomes dark purple from the right to the left.

DB: Yes.

GS: And it's the same for any other color.

DB: Right.

GS: Because it seems that the color changes also at the same time.

DB: It does. And that's just a visual perception.

GS: It's a visual perception. It's my fault.

CL: Well, yeah. We have not been able to find a nice way to display this in a continuous (span).

GS: This is very ??.

Andrew Davis: I'll just chime in right here. For a color blind person, this is not a good way of doing it at all. And you can fix it very simply by changing what's at your max, and what's at your min. Because the – well, of course, there are very many different kinds of color blindness, so I won't go into all that. But simply stated, the purple and the blue look exactly the same. So your max and your min look exactly the same.

DB: Yes. In other words, so changing the range of colors, or the selection of colors is –

AD: Yes. And I know that can be easily changed. So I would just simply put that in at the ground floor.

Brendan Griffin: Doug, I have another question on this. Isn't – your intensity factor is really a whiting factor on the original (count)?

DB: A brightening.

BG: It's a function of the number, that x and y in the original.

DB: Yes.

BG: So it's really a whitening factor. So you're always going to be very heavily (whited) towards the high intensity image.

DB: Correct.

BG: Have you thought of putting in a scaling factor on that whiting?

DB: Actually, in the software we do. So there are additional scaling factors that allow you to change the brightness independently of this. So there are ways to increase that, or reduce it. Essentially a contrast, brightness control. And that will allow you to vary it.

I think, if you recall, we're starting out with effectively two pieces of information, a ratio, and an intensity, and we're converting it to three pieces of information, RGB. So we're actually dropping – and that's the thing is that an HSI is actually a one to one, or a matrix operation that converts from red, green, blue to HSI. What we're doing is assigning ratios to hue, and the brightness, or the significance of the data to the intensity, and leaving the saturation at a constant value. So we're ignoring the saturation component in this.

XX: Could you calculate significance based on statistical certainty?

DB: You could, but it's not something we've done.

WL: That would apply a mathematical logic to the (contrast) you introduced in your previous example. The data that was black you essentially suppressed. But there is a logic for it,

because it's highly uncertain data.

DB: Right.

WL: Dividing by very low numbers.

DB: You could – well, actually, in the software I show up here that I'm using the intensity as a max of the numerator and denominator. In the software we give max, (man), mean, (sum, RMS). So there are lots of options there to select from to determine how you want to do the weighting. And it turns out to be a pretty trivial operation, from a software standpoint. So yes, we do give people different choices there, ways to do that.

So this is just the previous slides, using now showing both the gray level, and the HSI transform (Slide #14). And we do add the color bar to the bottom just so that people have a reference scale as to what colors correspond to what ratios.

DB: Did I miss something? (Too much light)?

XX: Yes.

CL: I would like to see ??.

DB: Not enough contrast. That was intentional, though.

GS: ?? contrast ?? yellowish ??.

XX: George, could you speak up?

GS: I was just wondering the meaning of the residual contrast on the yellowish part here.

DB: (Those are) high ratios.

GS: High ratios, and the ratios are (fluctate) – you retain some kind of fluctuations on the ratios?

DB: Yes. So there's still some noise associated with those calculations.

CL: We don't know this noise. These are RNA, and there is RNA distributed in the cytoplasm, which is the (rough endoplasmic reticulum). And in fact, I have (fiberblasts) where it's absolutely at the location of the andoplasmic reticulum.

Hojatollah Vali: Okay. Now the image we have down there at the corner, lower corner is, I think, C 12 and 14.

XX: Yes.

HV: Okay. In this case we don't really understand the relationship between the contrast. But if we would have a specific, let's say specific level (in) phosphorus, you would see that the two colors, the contrast is specific (for one) isotopes. In here now I see the membrane is blue,

whereas the nucleus is –

DB: ?? (regenerating here).

HV: Yes, exactly. And now if you would have the same composition both, would you get the same color? I mean, that's – because I would expect here, for example, the nucleus membrane should have the same color that's in the center. That would be – why not, (if there's) the same element?

CL: Not at all. That's a ratio.

HV: No, no. Imagine you have phosphorus, ?? phosphorus the brightest region in your image corresponding to one element. If you go ?? the color, you would expect the same. Now that is confusion, I mean, that is something really I don't understand.

DB: Well, –

HV: Why should it be two different colors?

CL: I'm not sure I understand your –

HV: No. ??.

DB: If you look carefully, the region up here is actually the nuclear membrane is much darker than it is in the region inside the nucleus, and definitely much darker than here. So there's this area that there's a reduced ratio crossing the nuclear membrane.

XX: ??.

DB: Why is it more prominent here? Because you have more counts. Remember, this has lots of counts, and this has very few counts.

CL: No we don't know the origin of ??. But again, when you do ?? isotope, as I have shown yesterday, you are always flat. It becomes – that's what you are seeing, flat.

WL: I think it's key here to remember that we're looking at – on the left we have elemental or isotopic images. What we're looking at here is a calculation. It's a ratio, so things are going to be different than what you might expect. Now the colors are not meant to indicate nothing else, other than a number, a result of a calculation.

And so one of the things, if you go back to your color block where you're trying to explain your – that one right there where you have ratio is sort of a y axis, intensity. To me, from a mathematical perspective, I like to be able to relate my calculation to a gray level or a color directly. And the weighting factor you have here in intensity, how is that – that's based on what here again?

DB: The number of counts.

WL: Number of counts in that pixel.

DB: Right.

WL: So it is a statistical weighting factor.

DB: Right.

WL: So then the brightness – your eye is brought then to data that is more reliable statistically. So this is a reasonable way to go. Except for the fact that, as you say, not everybody has the same color perception.

DB: Right.

WL: Now it looks like you're getting ready to leave color tables, and go on to other processing. Before you do that, have you considered using things like logarithmic displays? Are you familiar with the display methodology that has been developed by, say, David Bright at (MIST)?

DB: Mm-hmm.

WL: I like that a lot because when you have even higher (bit depth) data it really is flexible, and it's very intuitive, to me.

DB: Basically what you're suggesting is changing the scale. Instead of doing R min to R max, doing a log scale on the ratios. And that gives you much more – higher dynamic range, but it suppresses some of the noise and, in other words, now your display doesn't correspond to the noise levels of the images. You lose detail in a number of –

WL: Except that I think they corrected for that by using different colors for different logarithmic ??.

CL: Can you go to the next?

DB: Yes.

CL: There are several things I want to point out. One, here what we're looking at is an excess of one of the isotopes vis-à-vis of the other one. So it's truly to get excess of one isotope in some location in whatever we have (acquired). So the green is excess (nitrogen) 15 coming from the 15 ??. It's not at all random. It says there is an accumulation of 15 (in) signal at this level. So that's one information.

The second one about the intensity of images is what I have shown yesterday. We do not understand the formation of contrast, I mean, in detail. One can have an hypothesis. But I think that one answer to your question about phosphorus is that if we have two images that do not have excess of any isotopes, and if we take two closed masses, if we do the ratio as I have shown yesterday with 28 (cevisium), and 30, or with 12C and 13C, you may have images that are individually very contrasted, and the ratio doesn't display any contrast because it's equivalent, at every level of the (field).

Then there is the matter of logarithmic confirmation, and why we want to use HSI. Logarithmic transformation will never – and we have tried. We have plenty of transformation. In fact, in the IC program we look at them with 15 functions of transformation of color. We never have revealed the (dipplings) that I showed yesterday. These little areas in this enormous noise, if I may say, of excess nitrogen, we cannot see it. It's very flat, (these little) things. And the way we use HSI is that (effectively) we can play with all the variables so that we kind of make (appear) striking difference of color, not even using the (whole) ?? of ratios that there is in the image, but narrowing it, and seeing that at some place there is something. Then – and again, as I was insisting yesterday, we can draw around these pixels, and go back to the original images to touch, to extract the numerical information that have not been ??. So this is really to guide the analysis.

GS: Yes, it is not – what we discuss is here not to criticize, and to say it's bad work, but just – we wish to understand exactly what we see. And the point – you mentioned yesterday the silicon. But the silicon, okay, on the wafer, it is homogeneous. Here you never – on those images you never lose completely the intensity information. Because even if the isotopic ratio was completely the same, since the intensity (emission) on each point on the surface. So you will retain some structure that you don't know to what it is related. But it's there in the image as you get it.

CL: Yes, you have, again, variation of the intensity. But the ??, we control all the parameters. And so, for example, (all the absicia), the (world) of ratios that we want to look at, we could have fixed it at one, from zero to one. In which case what you would see is something which is entirely blue.

XX: I have a question back here, Claude.

XX: In every image, are you trying to have pixels which are going from the blue to the pink? I mean, it means in this image are some pixels which are pink.

DB: Right. Actually, for these images I autoscale, so I determine what the maximum ratio is in the data, and what the minimum ration is in the data, and allow it to scale automatically between those two points. So it's calculating (it). So yes, they're always – you can vary this. I'll show this later in the software. But for demonstration purposes here, actually some of the ones later I'll show that actually exaggerated it a bit. But I try to keep the data over the entire range of the color scales.

XX: Okay. Because I (cannot) see pink ??.

DB: There are probably one or two pink pixels in there, and you would be doing really well to see them, I think.

CL: And in fact, as I said, I mean, when we use it we play absolutely with all the parameters. So in this one, as I said, we could have blue, or if the green – (I say) the green, that's an area I want to see, so I will reduce the bandwidth, and I will put my maximum at 0.02. In which case the ?? will become purple. And I will effectively erase all the little dot of purple that you are seeing in some of the ??.

XX: We have another question back here.

BG: I'd just like to return to something that William raised. I think it's a function of color perception, and a previous statement from (Oxford) Instruments about five or six years ago gave a nice talk on – they selected colors based on the brain, or the eye/brain response. And so your color scale, in fact, for presentation should be based on that type of logic, because then you get a direct – more uniform perception, even in the cases of most men that can't see things clearly anyway.

So I might, just as a constructive comment supporting what William is saying, if you look at a color scale that is based on that brain response, it tends to avoid some of the issues that are being discussed here. (Because you have to see) the area to go back and take the numeric data out ??. You still have to be firstly aware of that being a significant area. If you don't see it, than ?? the color spectrum.

DB: I want to take a brief moment, and tell a story. You raised the question in a point earlier about people being color blind. As a post-doc I was at Baylor Medical School. And the Chairman of Medicine was Tony (Goda), and he, at one year, was President of the American Heart and Lung Association, and gave a keynote speech at a meeting for the association that year. And had asked me to put together a presentation for him showing some work we were doing using these different fluorescent dyes.

Well, I had red fluorescent dyes, and green fluorescent dyes, so all the images were red and green. And I neglected to go over this minor detail with him before. And as he made the presentation, I found out that he was red and green color blind. So it was somewhat of an embarrassing situation. So yes, it's something – being able to select ranges of colors is a very good point. And it's something that I'm sure any number of people – if you sample 20 people, you're going to find different perception of color in each of those.

Now I want to go on to another subject which, again, Frank talked about briefly yesterday. Actually, he just kind of casually mentioned this, but I wanted to go into some detail on this, primarily because this is a component of every software package, and it's a trap that many of you will fall into. And before you fall into it, I want you to think about what you're doing.

Noise reduction – one of the problems we have, and as you see in this previous image, there's lots of noise in here just based on the statistics, the (plots on) statistics of the (accounts) (Slide #15). There's information in here, biological information that, you know, high spatial frequencies, high detail in here that is not noise. And what you're trying to do is to pull out what's the (plots on) statistic noise, and save the spatial information. Right?

So there are a couple ways you can do this. One that Claude has been using, and it worked quite well, is a median filter (Slide #15). And a median filter, essentially you scan through an image, and for each pixel, you look at its neighboring pixels. You rank those, and replace the value at the center pixel with its neighbors, or the rank of its neighbor, the media rank of its neighbors. It retains sharp edges. So this will leave your high spatial detail in here, but it's nonlinear. So you have to be careful about doing operations which are nonlinear. As Frank

mentioned yesterday, it does not have a net sum zero effect. In other words, the mean of an image after a median filter is not going to be the same as the mean of an image before a median filter.

Now convolution is simply weighting the neighboring pixels by using a mean, or some kind of a weighted kernel (Slide #15). You know, so you select some matrix of weights in the area around the pixel, and you sum those together, and divide by the sum. I'll show this in a little more detail here.

Convolutions are – they do have a net sum zero. And actually I think all the images that Frank showed – the ratio images that Frank showed yesterday were all convolved. So they were blurred intentionally. Now the advantage is that you get this square root of N noise reduction. So if you average 9 pixels together, the noise goes down by a factor of three. If you were to average 16 pixels, the noise would be reduced by a factor of four. So you get – you reduce the noise –

XX: It depends on the number, (on the intensity of each pixel)?

DB: No, it does not.

XX: It's just a sampling.

DB: It's just a sampling.

XX: ??.

CL: ??.

XX: ?? if the ?? have the (same intensity), yes. (Then you get a factor of 3).

DB: But it's the relative standard deviation of –

XX: ??.

DB: It's linear, but it blurs the image, reduces the spatial details. Just quickly, again, to show what a median filter – if you were to start with (a block of) pixels like this, and as you would probably – frequently you observe, you've got some outlier, you've got some really bright pixel in the center which should not be there. It's some statistical anomaly. By taking a median you just scan – you take all nine pixels, sort them from smallest to largest, and find the one that's in the center, and replace the center pixel with that value. Is that clear? (Slide #16)

So here are a pair of images – so this is the (mass) 27, and its corresponding HIS (Slide #17). And on the right you see the median filter version of this, and the median filter – and the ratio as a result of calculating with this median filtered image.

So you see there's clearly lots of spatial – there's a lot of detail in here now that appears that wasn't there before, or is obscured because of the random color components of the field before.

And just to show you convolution (Slide #18). In this case I've chosen just a two by two convolution kernel, which is essentially averaging each pixel with its four neighbors. This can increase to any size you want. So there's no –

DB: It can be as large as you'd like. And, again, you multiply these times the corresponding locations here. Sum those together, divide by the weights of the coefficients, and replace this pixel with that. And on the next slide you see the same thing, but now you'll see that this one is clearly blurred, compared to the original data. And there is also some loss of information in the HSI image, or the ratio image, as compared to the original data (Slide #19). Yes?

CL: Something I had not noticed, but as (Brenda) saw yesterday, there is a ripple from the (cutting), and they appeared in the (converted) image. We see them. We see wavy (intensity) in diagonal. There are lines that are slightly (whiter). And this is – that's funny, because it's coming from the (wave of the cutting) that we see. So that's a different example of whatever, of ?? in this.

DB: That's the screen, I think, Claude.

CL: No, no. It's not the screen. It's not the screen. Absolutely not. And we have it on the HSI image on the right. You can do what you want. These are the (parallel) ripples (of the slice).

DB: The advantage of convolving – and I don't mean to pooh-pooh or to dispel ??. You're effectively – by averaging, you're increasing the spot size of the system, and you're getting more counts. You can think of it in terms of you're grouping more counts into a single pixel. Trying to maintain the spatial relationship, but you're – go ahead, George?

GS: Yes, I think you are right. But if it depends now of the sampling when you take your image. For instance, if you have an image with a (probe) which is .1 nanometers over 5 microns, and with 256 pixels – 256 per 256 pixels, then the convolved image will not be so bad. You agree?

XX: Yes.

GS: Because the sampling will be appropriate for the convolution.

DB: Now, I want to go over another part of convolution is this thing known as high pass filtering (Slide #20). And in a high pass filter, you'll notice here, instead of summing the guys together, I'm subtracting the neighborhoods, the neighboring pixels. And so you replace the center pixel with – by weighting it by some value, and subtracting the neighboring pixels. The result of this is an image like this (Slide #21).

Now this does not work on images which have few counts to begin with. But if you have an image which has lots of counts. In other words, the noise is relatively low, then you can start with something like this, and end up with something like this (Slide #22).

XX: ?? sharpening ??.

DB: The sharpening, right. Now, the question is, when you do things like this, when are you making data? Go ahead, George.

AD: ?? about (seeing), it's not about data. Ultimately, this isn't about data, this is about (seeing), this is about presentation.

GS: The data is here. (It shows the) things that ??. Then you can go back ??. So it's an interesting ??. And again, it (depends on the) ??.

CL: (I have said fifteen) times I use this to guide my viewing. That's all. That's the (essential) point, to guide the viewing, to guide the ??. And so that's good.

Francois Hillion: I think we can see the membrane, the nuclear membrane there on the right (Slide #22), which is absolutely not visible there on the left.

CL: Absolutely.

BG: Is that because it's pink we can't see it against the –

DB: I'm sorry?

BG: On the bottom right, is the membrane purple?

XX: It's blue.

CL: We keep the intensity, and it's a more intense blue.

HV: Do you have the original ratio? Where's the original ratio? Not the colored one. Why is the original black and white ??. Then we could compare this image here.

DB: We'll do that when we go to – when I do the Image J. Remind me. We'll come back and try that.

WL: Just one last comment. The convolutions you talk about, these are very useful. You always have to go back to the original data to calculate a result. But doing a convolution like this can help you identify phases that did not seem obvious to you in the original data. Because, like you say, noise issues, for example. What this does is allow you to look for relationships between adjacent pixels that your eye did not pick out.

And so you are also correct, though, that if you use a filter, a convolution kernel that is too severe, you will introduce artifacts, and "phases will appear," I'll put in quotes, that really don't exist. But if you go back to the original data, it will all – you can always check yourself. So you can use it for display purposes, and for conveying a message, but you can also use it for doing – as you say, as a guide.

CL: Yes, that's (essentially) it's a guide for identifying a region of interest.

XX: And then calculating a result.

DB: Very briefly, if you were to take a cross-section through an image, and you had a single pixel that had, let's say, for the sake of argument, 1000 counts, and all the neighboring pixels were 0. So let's assume we've got some structure, some feature in the microscope that we know all the information is in something less than a single pixel. Ideally, if you looked at the profile of that you would see 1 pixel that had all the signal, and everybody else was 0 (beside it). Right?

In reality, what you see is something that looks like this. And I'm going to exaggerate it. I'm sure these microscopes aren't that bad. In other words, there is some overlap, there's some signal being spilled out from this guy into the neighboring pixels. Everybody understand that? It's a point spread function. And if you think of this in two-dimensional space, you've got a circle.

And I think the point Frank made yesterday was that he intentionally samples. His pixels are much less than the size of the (beam). So he is intentionally convolving, or blurring the images. And he justifies – and rightfully so – that he can reduce the noise afterwards by doing some convolution operation. Right?

So what we're doing mathematically with a convolution is what the microscope does, depending on the size of the beam, and the size of your sampling area. Right? So the microscope convolves the data. What you start with, the microscope is always going to convolve it. And what you're trying to do is to find out how much of it was blurred due to the microscope, and how much information? What was the original data there? So when you start looking for high detail, you've got to think in terms of, "Okay, what's the blurring function of the microscope?"

Now the point George was making, and I wanted to make as well, if you know this function, you can mathematically undo it. And that's a deconvolution. And a high-pass filter, what I've just shown here, is kind of flipping a coin, and guessing what the shape of that kernel was. So you're making a guess as to what the data should be. And sometimes, you know, you can guess pretty good. But be aware when you start doing these convolution or high pass sharpening filters, it's – you know, you're in Photoshop. It's art; it's not science.

If you go in and physically measure this, then you've got some a priori information that you can go back and do a deconvolution, or do a sharpening that will give you information that you probably wouldn't have seen to begin with.

GS: I agree with what you have said. If we can go to the previous slide you showed with the numbers (Slide #20). Yes, it means that, for instance, on the left, when you have 25 here, if it is a real detail it might just be that the 9 and the 5, it's because the probe, the size of the probe and the position was just touching the detail, and giving 9 where there is 0.

So what would be nice is to take a detail, and register the detail with different sampling, if you are really interested, and then look at what is the most appropriate ?? (to) – if that is the aim, the objective.

CL: Unfortunately, a detail means that it's not general. So it's difficult to sample the detail several times.

XX: ??.

CL: No, no. In our case.

GS: The other point I want to make is that here it's seen in an abstract way, but when you see the membrane, you have the same detail which is repeating everywhere. So you have a structure. And when you go to two dimensions, you add something to the understanding of your image.

BG: I agree with all of this, and I think it's extremely important. (A) question, perhaps,?? Is that we're looking at ratios, and we're looking and interested in changes of ratios. Is it useful to look at a differential of the image, where you're looking at ?? change (of adjacent) data points?

DB: Yes. So you're looking at, essentially, a derivative, or doing a first –

BG: Because you'll take out then, for example, the areas where there is no change.

DB: That's typically what's known as an (edge) detection. So there are these operators that allow you to do things like – yes, edge detection. So they show you areas that will give you contrast where you have changes, relative changes in brightness. Now, we've covered this one. Okay.

Here I've stretched out so the pink is clear with everyone. And I wanted just to review. We have the original data in the upper right. We have a median filter – I mean upper left. We have the median filter in the upper right. The convolved, blurred image in the lower left. And the high pass image in the lower right. So you can see – and it doesn't take much to see that clearly you can see details in this one that aren't in here. You can see details in here that aren't in here. And the question is, if you look in this, and go back to the original data do you still see the detail?

And here's just a zoomed up area, again, just showing the same four regions (Slide #23). So the original data, the median filter. You can see there's some structure in here that is somewhat blurred, and these guys. Specifically, there's a pixel here that disappears in this blurred one. And there's other structures in here that are retained in this one, but lost in that. So depending on which operation you do, and if you're just randomly doing operations like this you'll get different data. I think that's in my conclusion here (Slide #24).

So HSI improves the ability to visualize details in ratio images, just because of the noise suppression, and weighting of the intensities. Median filters reduce the noise with a minimum loss of detail that can be used to select regions of interest for analysis. They cannot, and should not, be used for analysis, because it's a nonlinear operation.

Convolution reduces the noise, but reduces spatial detail. But it is a linear operation, and it can be used for analysis.

CL: Ugh.

DB: Frank would disagree with you on that.

CL: No, it could be microscopy we don't have the detailed things. There we have the pixel by pixel number of counts you use them. When you get your mean, in some ways you are doing a convolution.

DB: Right, exactly. And high pass filters can increase the spatial resolution, given that you have some a priori information of the system's point spread function. Now it's 10:00. How much time do I have, Claude?

CL: Whatever you want.

DB: Alright. I want to switch over to – I was going to do a quick demo with Image J. But I wanted to cover one thing, for those of you using other software packages. Not that you have to use mine, but I want you to think (about it). Cameca, when they generate an image file, goes through a great deal of detail to add a lot of information to that image file (Slide #25). And Francois, you're nodding so you can – I'm only (getting) a small fraction of the information up here, but I wanted you to understand there's a lot more detail in there than what I'm presenting here. That's free. I mean, that data is collected and recorded automatically for you. So if you use any software package that converts an image file, the Cameca image file to another format, you should be aware that you're probably throwing something away. And as a general rule, scientists are really unhappy about throwing away data. So I wanted to make a note about this. In the software we've developed, yes, we do preserve this information.

(End of Slide Show – Beginning of demo with ImageJ, which can be downloaded at: http://www.nrims.hms.harvard.edu/software.php)

CL: Yes, and for us it has become very practice, and very important point also ??, which is to give an (identifying) stamp on each of the images. Because very quickly we ??, and it comes from them, and we have no idea where they come from, (when) have been acquired, what is the size of the field, etc. And to carry along all the image's original information is something important.

XX: ??.

DB: Actually, no. It's Washington, Seattle. Rainier. Flying with a digital camera looking out the window. There are a couple of ??. Okay. Very briefly I want to go over a few things about Image J. The Image J, as I mentioned before, was developed at NIH, and actually it's an offshoot of a software package called NIH Image that's been around for 10, 15 years, a long time, and has continually grown.

The problem with NIH Image was that it was originally written for Mac. And although I know there are a lot of Mac users here, and I don't mean to offend anyone, but there are a lot of people out there who don't use Macs. So to get around that problem, he transferred all the software – Wayne (Rashall), who developed it transferred all the software to a Java application. Java is a scripting language that is – or not a scripting language. Actually, it's an interpreter that allows you to compile a program that will run on a Mac, Windows, Linux, Unix, practically everything. So the nice thing about Java is that literally the same software – you can drop the

same software from a Mac system onto a Windows system, and continue working. So it's not recompiled every time you go from one system to the other. So for those who have Windows, and Linux, and Mac systems, it's a nice application, because you can share it between a lot of different systems.

It also has kind of an interesting feature: it runs across the web. So you can actually run Image J inside of a net browser. I don't know what practical reason – actually, Claude and I have talked about being able to present data from the facility, or exporting or sending data to the facility. So there may be some practical applications of being able to run an image.

WL: Yeah, I deal with people all over the world in my company, and they can't each load and install stuff on their PC, or whatever machine they're using. So having a web browser's great.

DB: So it's a nice convenient package from that standpoint. It's quite (mature) for what it does. And considering the price, it's an extremely well-designed, well thought out application. It's free. And the website that I gave on the first slide, that actually – if you go to that website, click on download, and pick up whichever version you'd like, and it's easy to install. I think probably anyone in here could have it up and running in a matter of a few minutes, maybe with the exception of Claude. But it's easy. Go ahead.

WL: Well, I can ask Francois too. The ?? files are in Unix format, are they not?

XX: It's not a Unix format, it's ??. It's just a format. It's a Cameca format. It's independent of the operating system.

[Inaudible simultaneous comments]

DB: Just so everyone – Claude, we haven't figured out how we're going to distribute this yet, but there's a plug-in. Plug-ins – Image J has this ability, there's a folder called plug-ins that you just drop another folder in there, and it automatically shows up on this menu. So I dropped a folder named NRIMS, (with a) facility here named (open) ??. So these are raw Cameca file formats. And clicking on "open." Did I hit open? No, I didn't. Boom.

So this is an application that will read the Cameca formats directly. And it's smart enough to know whether it's Big (Indian), or Little (Indian), or Unix, or whatever. So it will translate between all the different types. So this is a (free) file format. Cameca images can – depending on how many masses you measure simultaneously, that's how many images will show up in a single file. So it will automatically pop all the different images up at once.

AD: Will you deal with multiple ??.

DB: Yes. There's – I can show this. There's a utility that lets you stack images.

AD: But a single file can't have multiple images ??.

DB: At the same plane?

AD: Yeah. ??.

DB: The file format –

XX: It's all one file.

DB: I haven't placed any restrictions on how many images are in the file. So how many images were recorded; that's how many it will pull out. And if they're time-based, or mass-based, or whatever, I assume the information is in the attributes to resolve that. Which I wanted to – was another point, before I forget it, and I skipped over earlier. What Frank suggested yesterday about doing time series, from what little I understand about the way Claude operates here, I don't know that that's practical on a day to day basis to be able to do a ten hour acquisition of the same image.

But if you are looking for detail – going back to what we were talking about earlier about is there some spatial information, is there some spatial detail there or not? If you have repeated images, and you see the same structure each time, then there's probably a good chance that that is something real. If it's fluctuating and disappears from frame to frame, then it's probably not. It's complicated by the fact that you're – this is a destructive operation, so you're changing the sample as you measure it. So I don't know that that really – you know, assuming you're only etching a small fraction of the structure, then you may, in fact, be able to see it with repeated measurements. But historically personal experiences, any time you can do multiple images of the same structure, or same feature, there's a lot more information there, and it's well worth the trouble if you can afford to do that.

Now very quickly I wanted to show – so (they) designed this in a way that hopefully we can add additional features to this. And my intent for what we're doing with Image J here is something to – you present this to the community in general, Cameca specifically, and see what – who else can contribute to it. So Frank made the comment that it was open source – and I don't mean to sound like, you know, it has some bad connotation. From a business side of this, with ten users, whatever, five users we have in the field, there's not enough of you out there for someone to be developing software for this as a business venture. You have applications, you have needs that there's no commercial incentive for someone to go out and try to address those. And just as kind of an editorial comment, I think the best we can hope for is to pick a platform, any platform, and standardize on that, and let people contribute to it, and share as much of it as you can.

WL: I think I would add to that just that now that you have a format converter, essentially, ?? in the files, and (write them out), that's the key step. Then people can take whatever package they want ??.

DB: That was my next point was that the (save) function that I've added here actually goes out and just immediately dumps it back in a TIF file, in a TIF format with all the attributes saved with the images. So all the metadata saved with the images. So the worst – the least you can do with this, if you wanted to go from your raw data to Photoshop, open it, save it, close it, and open Photoshop. TIF tag, the image file format, is a format that Photoshop, and practically everything out there reads.

CL: In a 16 bit.

DB: In a 16 bit with the original data.

CL: All the (quantitation) is kept.

WL: The only thing with TIFF is you will find that not every TIF reader's the same. It does vary. Even though it's highly specific, and it's a well-defined format, not everybody reads it correctly.

DB: It has this ability for people to get format tags, and not everyone reads those tags the same way. Some people don't even publish them. So it is pretty – and 16 bit, if you save it as a 16 bit TIF, Windows will not open it. Actually, I think Mac will, but – I'm pretty confident a Mac will in fact open it. But Windows applications will not open it.

XX: Have you actually tested your file (reader) on stacked images from (SIMS) acquisitions?

DB: No. And if you would like to send a stack to me, I'd be more than happy to give that a shot.

XX: I can give you ??, which should be the same format?

DB: Yes, that's what I understand. We're on the same format.

XX: How do we measure in practice the (size of the beam) on the machine?

FH: (You can measure it).

XX: How?

FH: Just you find the data in the image, sharpen up. Maybe I can draw something. (It's exactly) the same image you have – same (graph) you have on the board there. ?? the data is very small compared to the (prop) size. And when you do a scan in the image, what you will see is the (prop) size. Or the data is larger than the prop size, and what you will see is (datase), instead of the prop size. So of course you have to find in an image a very small (datase). ?? that is very small, or small enough that you are able to see when you do a ?? scanning the image the prop size. That's a way we measure the prop size when we do specification on the instrument.

CL: ??.

DB: In theory, because this is a destructive process, if you start out with a perfect surface, and just position the beam – and I'm reasonably confident. Can't you etch a point spread function – the inverse of a point spread function?

FH: What you have to get is a sample like that. (C) ?? here, and no (C) ?? there. (C) can be whatever it is, (it doesn't matter). So you just scan the beam there, and you see on your image something like that ?? (border). Then you do a ?? scan on your image, and you see something

like that. And if it is sharp enough, (if you have just a) very, very sharp edge, what you see here is your prop size. Of course, it's relatively difficult to find such a sample. ?? sure when you have a (silicon carbide grain), for example, with a sharp edge, it's not easy to find. We have some ??. We have a very, very nice titanium carbide sample. On that kind of sample we are going to measure prop size down to (2500). There are many samples ??. ?? the sample itself. (That's) the way we measure that. It is the only way to measure it.

XX: And that's – maybe could you comment, Francois. If you have a heterogeneous sample –

CL: May we reserve questions for tomorrow, maybe? I mean, (we're) drifting later.

DB: We're trying to finish up here, and (Cliff) still has some time to – where was I? The other thing, I just wanted to show this. When you go to the Image J website, all of these items on this menu, the plug-in menu, these are all things that people have contributed to the application. And, for example, there are ones to do deconvolution. Which if you ever get sophisticated enough to do this, it's a good place to start. Feature extractors, image correlator. This was the subject, I think, that Frank was talking about yesterday of being able to measure shift. So there's a utility here for doing image correlation so you can measure the displacement between pairs of images. There's an advanced IO operator which reads a lot more file formats, and so forth.

My point is that it's very easy – they've designed this in such a way that it's very easy for people to contribute to this, and there are, I don't know, probably hundreds of plug-ins now available for Image J that do all kinds of really nice imaging tasks.

WL: Sorry to ask, but I don't know how they – how do they umpire whether or not – referee such code? That's an issue.

DB: Actually, it's in the Source (Forge), and there's someone who manages it now. So there's a website.

WL: So they run – they corroborate the calculations? That sometimes is something to be aware of.

DB: You know, Java, unlike C code, and standard operating software, Java will – if it will crash, Java won't even compile. So it's –

WL: I'm not worried about crashing. I'm worried about (correct numerical) results.

DB: I (want to say) caveat emptor. Don't assume, don't trust anything. When they tell you that it does something, go out and measure it. Don't be naive to think that because of this reason, because somebody may have put something in there that is numerically incorrect. But as best I can tell, the things that I've done with it, it's very good.

Now, just to go very quickly over the – there were a couple other buttons up there on the top. Give a brief explanation. If you close an image, you can go back and reopen it. So that's what the open button is for. I mentioned the save actually just dumps all the data in a TIF format to the original folder.

If you were to select an image, and – I can drive this guy now – do something like a (calcium blur). And say, okay, convolve that now. Which is pretty snappy if you understand. If you have altered the data in a way that you're not happy with, then the restore function brings it back. So that was one of – another key feature. And I assume people would want to play with a lot of functions, so I wanted some method of being able to go back to the original Cameca file, and pull the data back in.

AD: Just ?? on that, that takes you back completely to the original file? So if you do a number of steps, it just takes you – it's not a single step back?

DB: There is, in fact, an undo function here which I think stores, I don't know, maybe two or three steps. It's a memory thing as to how many steps it does. But it – so you can undo single step operations.

Now just to go over how we were doing these mass HSI images. Here is a (panel). You select which mass you want as the numerator, and the mass for the denominator. You can choose a ratio. So remember my R min, R max? You can select a range that defaults to some big number, like 25. This (alter) range button actually goes out and looks at the data, and automatically calculates what the range of ratios are. You can select a threshold for the numerator and denominator. And I use this in cases where – I mean, if you've got pixels out here where you only have one count or two counts, there's no point in trying to calculate a ratio from one or two. The signal to noise is the square root of the number of counts. So by default I set it to 3. You may want to choose a smaller or a larger number, but it's just – again, it's one of these gives you the ability to select some number of counts, and say, "Anything below that, I'm not interested in." So just make those zero to begin with.

The RGB max and min. This was someone asked earlier about adjusting the contrast of the colors. This allows – by default it uses a full range. And the transparency – remember I was telling you about how we calculate the intensity? So you can use either the brightest of the numerator or denominator, or just the numerator, denominator, the minimum, the mean, the sum, the RMS. And if you want any more – I thought that was probably a pretty good selection to start with. And finally, there's this little scheme for the color bar across the bottom as to whether you want (it off), or just a bar, or the labels. So if you click on the HSI button, that's it.

And just, very briefly, to show you the effect of reducing the max (in) the color space. So if I do the same thing again so I can effectively brighten the image compared to where it was before.

WL: How about a micron bar?

DB: Good question. There is somewhere on here. Scale bar, like that. Okay?

XX: Good question. Good answer.

DB: That's why I like it. That's why I chose this package is because there are a lot of features like that that are already built in. Somebody had thought it through.

AD: I just wanted to ask the color blind question. Can we change the ?? colors?

XX: Color blindness is an issue.

DB: Yes.

AD: Is it ?? right now to?

DB: It's fixed, currently, but, I mean, there's one function that does the color, the hue. (Cosine), whatever function that does the – so I think what's reasonable is to probably come up with a series of color scales that people could choose from. And still apply the intensity modulation to it, but just let them choose what standard color scale they want.

WL: Can't you just ?? table now, using the top level menu?

DB: Yes. Actually, these are RGB, so in theory you can convert it to an 8 bit color, and then you can go in and change the color scale to (fire), whatever that means.

XX: How about gray?

DB: Yeah.

FH: Just as an example, can you do (original) ??. (Say) something about that?

DB: Sure. So this one we'll leave in. So there are these little icons across the tool bar up here that allow you to draw regions of interest, so we'll pick one here. The problem that – and if you're like me, and you can't draw that well, I think this will do it.

The problem that, and one of the shortcomings of Image J is that it only – you can only draw one region at a time. Now to get around that, somebody after the fact came up with this obscure little tool, hidden under "analyze (roy) manager," that let's you add regions to some list. So I can go over here now, and draw another region around this guy, and add that. And another region around this guy, and add that.

So if I go back to my original denominator, all I have to do is select these different regions, and they will appear as overlays in that data set. And there's a measure function here that allows you to collect the mean, the area, the min and the max from all the selected regions of interest. So, yes, you can do that.

XX: ??.

DB: Hey, like I said, the price is right.

Peta Clode: Can you export the results file (still)?

DB: Yes, actually – I think it saves it as a tab-separated – yeah, just as a tab-separated text file. But it's easy to import those into Excel, or other applications.

XX: Doug?

DB: Yes?

XX: Going back to the region of interest capability. Do you also have a histogram function that you could do on those region of interest? Because sometimes that shows features and phases.

DB: There is an analyze histogram, but I don't think the histogram works on the region of interest. I think that works on the entire image.

XX: (Does the real) number floating point ??.

DB: Yes. So all the ratios that I'm calculating in Image J are actually saved as floating point; 4 bytes per pixel.

WL: So the histogram will display a floating point?

DB: Yes, it will. So it does handle floating point values, from that standpoint.

XX: The advantage of the region of interest (zoom) on the histogram is that ?? image you can just (press any) ?? (small features) ??.

DB: Well, the measure – the histogram, when I was doing these measurements over here, those are, in fact, the integrated intensities from those regions of interest. So they're only from the areas that I've drawn. And so you don't have to do – I mean, the histogram is calculating the same thing as those are, it's just that that's just giving you the raw data. Well, I'll rephrase that. The histogram is actually calculating the data from the entire image. And there probably is a function in here to do a histogram of a region of interest only, but it's not one that I've found.

I wanted to make – Claude, it's 10:20. Do you want to save the IC demo, or do you – how do you want to do that?

CL: I would say if you feel like continuing, you may continue. We began late. We started late.

DB: Okay. Well, I wanted to make a comment about Image J, and it's a subject I know Claude's pretty sensitive to. Applications like this, if you are dealing with a dozen, or a couple of dozen images, this is great. It's a really nice application. If you are in a production facility where you're trying to analyze a hundred, or a thousand, or a couple of thousand images, or a couple of hundred images with tens of regions of interest per image, this is a pretty cumbersome way to go. And to say that, you know, yes, if you want to park a graduate student in front of a computer, and tie them to it, and let them stay there for weeks at a time, great. But I wouldn't – you know, I'd have some question about the competence of the results, you know, if somebody is going through doing that kind of grueling work day in and day out.

So one of the questions when you start looking at software for a facility is how much are you going to be doing? And if you're doing this infrequently or only occasionally, this is great. This is really nice. If you're doing it, as I know Claude does, day in and day out, it has its limitations. And as to how much of a production operation – I mean, it's an open source, so if you want to add software to it to turn it into a production operation, great. But it's not something that I would – it's probably not the best model, as far as the software design, for handling large

volumes of images, and large volumes of data.

XX: The thing with open source is (anything) you develop you're supposed to also share.

DB: Right. Assuming you –

CL: ??.

XX: No, that's a good thing. It's an issue, though, for a company. ?? (a product) or a calculation, sometimes I can't use those things (for a) ?? reason.

DB: You have to, yeah, give that some weight.

XX: Is it possible to ?? go back into the image ?? (taking all the) ratio ??.

DB: Let's clean up some of the – save, clear. So we're starting with –

XX: ?? ratio images.

DB: Yeah. I want to see if I can find the one that I just... Now so we're going to ratio – let's say that same guy, so we'll just calculate. So this is the raw ratio, and these are floating point data. So you can see in the upper right corner there on the little status bar, as I mouse around in here.

DB: – (32) bit floating point. There is some method of binary (threshold). And I don't know if this is automatic, or – well, I did that one. Yeah, that's the control-Z function. So there are some methods for doing that, but it's not a –

XX: ??.

DB: Let's see. Window level, yeah. There you go. It's probably – so you can go in and select some region in one of those peaks.

GS: I want to remove ??.

XX: (Then you want a different transform) ??.

DB: Yeah. There's a thresholding. There's probably a thresholding. I can show you that in IC, but I can't show you this here.

XX: You can do it with ??.

XX: Does your plug-in work with other programs? Your file format, it is a Photoshop-compatible plug-in?

DB: No, these are all Java. So this will not – you can't transport this. Unless they – no, I'm pretty sure not.

XX: ??.

CL: Okay. I think I'd like that we move on a little.

DB: Okay. I think this is probably a good place to stop. And I was going to show just briefly – so this is the IC application. And as you're up in the lab later today I think Claude has some other things that he's – stuff that he will be showing that uses this as well. But we did develop originally – we developed the HSI as a plug-in for IC, and you'll see that it has pretty much the same properties, capabilities that we have with the Image J. The one thing that we've done with IC that we did not do with Image J is the utility that allows us to load a stack of images, and treat those as groups of series. And it automatically records.

So this application does, in fact, let you do multiple regions of interest. So you can draw regions of interest, and record those with the image file format, and bring them back at a later date; unlike in Image J where it has a rather cumbersome approach to handling – being able to do regions of interest.

And there's a function in here that allows you to – when you analyze what it does is it goes through all the regions of interest, and all the images in that stack. And Claude, I think in this case I was probably using numerator/denominator ratio, or one of the ratios. But I think you mentioned yesterday you do four masses, and two ratios in the HSI. So a total of seven images per series. And what this enables us to do is someone can sit down and select areas that they're interested in, and save those, and then come back later, and it's more or less a single button operation to go through and process a large data set, or large numbers of data over a short period of time.

And again, the point I was making earlier was when you're in a production environment this is what – this is something you have to consider as to how much time it's going to take, and what your throughput needs are. This is not an inexpensive software package. It's in the $6-10,000 range. But it's, for the kinds of things for people who are doing this on a routine basis, it pays for itself pretty quickly. Any other questions?

I wanted to show one last thing. One of the things that Frank mentioned yesterday, and I wanted to – IC is what's known as a data flow. And for those who have used Labview, it's essentially you have icons that represent your data, and icons that represent your functions, and you connect the dots. So there are a number of applications on the market now. But I had an application that we developed some years back which is a similar problem to the one Frank was talking about yesterday. If I can find the data set here. This is a synthesized, shifting data set. So what I did was actually I have an image here that I rotated over, you know, just a series of images, and rotated those.

The problem Frank was describing yesterday was this thing of drift. The images shifting over time. And the correlation method that he uses, which is the same method that this application uses, the one subtle difference is that this guy actually does a subpixel registration. So you can determine how far things have moved to a fraction of a pixel.

Back to the question of the point spread function. What Frank was suggesting is that if a single pixel is much smaller than the point spread function, then a pixel shift is of minimum consequence. When you start doing (auto) correlation methods, and looking at translation (in)

images, you can actually very easily detect a tenth of a pixel shift in an image. So I would think you actually – you probably can, if you're doing whole pixel shifts, you probably are blurring out the data more than you'd need to.

Just to show you how this guy works, it has a little target, so it's kind of like a missile targeting system that you pick an object that you're interested in tracking. So you put a little box around the object, and it gives you this correlation image of what that looks like. And then you say, "Go," and it will lock on to it, and track it around.

So you can do this, actually, if you're looking at a problem where you've got shift in an image, you know, spatial shifts in an image, doing a simple registration, and moving it left and right will sometimes fix the problem, sometimes you may actually do more damage than good. And you want to be careful about when you start doing some of these iterative processes, you can end up damaging the data, or reducing the data, in a way, or adding noise to the data, in a way, that you would have been better off just to have lived with the problem to begin with.

Being able to handle things that are shifting laterally is pretty trivial. If you start getting into rotational components, then you need a little more sophisticated software package that will handle not only shifting an image in lateral dimensions, but being able to rotate it as well to be able to keep it centered or registered.

FH: Yes, but you can have a problem if you follow just (a datase). That will be (sputtered) away during time. (So) you are completely lost.

DB: The point that he made, which is a good point, this is a very sensitive algorithm. Well, I'll rephrase that. It's very insensitive to changes from one frame to the next. In other words, the structure can change. Statistically what you're doing is you're modeling – you're finding a pattern, and trying to lock on to that pattern between a series of frames. The pattern itself will change. So the gain of the lock may change, but the actual position of it is very –

FH: ?? disappear completely.

DB: If it disappears, you're lost.

FH: That's why doing auto correlation of the whole image is, I think, something better. Because a data ?? (ten hours) ?? with (forty) ??, you can have a data that completely disappears.

WL: The other thing, I mean, (you might want the ability to change your reference). So you may not always stop with the (top image) as the reference ??.

DB: Right.

XX: ??.

GS: If you extend ??.

DB: But it's fundamentally a tracking problem, and there are probably as many different tracking algorithms out there as there are people running them, you know, the software to use

them. So it's something that has been addressed by a number of places. Okay? And I think this is probably a good place to stop and let everyone have something to drink. Thank you.