All Hands Kick-off (2024-07-26 08:40 GMT-4) - Transcript

Attendees

Anastasia Yendiki, Andras Lasso, Andras Lasso's Presentation, Ben Murray, Ben Murray's Presentation, Bradley Lowekamp (NIH/NIAID Contractor), Chai Kaidong, Chaitanya Kolluru, Davis Bennett, Hastings Greer, Hastings Greer's Presentation, Iaroslavna Vasylieva, Igor Tatarnikov, Jeffrey S, Jennie Wu, Jiaji Chen, Jiaji Chen's Presentation, Joel Lüthi, John Bogovic, John Bogovic's Presentation, Josh Moore, Junxiang Xu, Luca Marconato, Luca Marconato's Presentation, Mahmood Mohammadi, Marc Niethammer, Marvin Albert, Matt McCormick, Matt McCormick's Presentation, Matthew McCormick, mohamad hawchar, Nick Tustison, Nick Tustison's Presentation, Niels Dekker, Ruben Dries, Scott Gerfen, Shang Mu, shruti varade, Stephan Saalfeld, Stephen Aylward, Tao Wen, Tom Birdsong US, Veronica Jarzabek, Wouter-Michiel Adrien Maria Vierdag, Wouter-Michiel Adrien Maria Vierdag's Presentation, Xiaoman Wang, Yael Balbastre, Yurii Zubov, Ziquan Wei

Transcript

This editable transcript was computer generated and might contain errors. People can also change the text after it was created.

Matt McCormick: there is also a chat, but I'd prefer to use the zulub chat instead possible. You can see everyone on the call here, too. Any questions regarding zulub or images the island or Google meet

Matt McCormick: so yeah, this is Organizes this hackathon a grant from the NIH. So thank you to the NIH.

Matt McCormick: The agenda now, I guess we have maybe 20. Minutes before we start on the tutorials. These are meant to be open discussions really from different folks who have on What they're using? For special coordinate Transformations and how they're using them and what their applications are. So first, we have a tutorial from the geodo. folks so George and Ruben from Boston University will be sharing how spatial transforms work in geodo, which is a spatial transcriptomics analysis tool and then we're going to be joined by Murray. From Kings called London to talk about benai.

Matt McCormick: lazy transforms and the geometric transforms proposal So when I is a deep learning.

Matt McCormick: library for medical imaging develops mostly by Nvidia and in Scotland and a larger community. And so hopefully talking about how lazy transforms work. in that case we have this idea which

Matt McCormick: The only transform on these are transforms help with and something like in ik we have this idea of composite transforms. So the idea of you have multiple simpler parametric transforms that can arise from different stages of automated registration manual registration and you want to compose them into one mathematical objects that runs through the transforms in a composite chain rule type way and

Matt McCormick: Doing that. in one operation avoids a lot of the description errors that you get when you maybe apply to an image and then have to interpolate and then apply it again interpolate. So that's what the lazy transforms are. And then in one eye there. all means ours evolving to get Geometric coordinate transform support and there's basic affine transforms and they're working on more transforms. We learn more about that and that section. And then Luca who's here with us? Is going to be talking. more with a lot of experience on coordinate Transformations and how they work in spatial data, which is another space transcriptomics Plus image data plus all the

Matt McCormick: interesting spatial data types and his experiences and what he's implemented there and how we can work with that closely. after that I don't have it listed here. But wood or Michelle our Transformers proposal is also involved with spatial data is of nepari. We'll talk about how that works in the party. and then John will talk about How it's works with?

Matt McCormick: Big work as he talked about the Java tool for registration. Then we'll have Nick tustison from ants talk about. How speech Transformations are used for most brain? mapping and how a more advanced application of defining a transformation that

00:05:00

Matt McCormick: defines the relationship over time and over developmental period of the brain I can use that and apply that and create that. And in the afternoon, we'll have Hastings Greer from UNC talk about how space transforms work in his registration tool that's made. to do deep learning base registration

Matt McCormick: between all types of objects or to jam any other description of the registration method there? so

Matt McCormick: this is mostly delicious mostly images. probably could extended. Yes. We'll see so mostly images though. So that's an overview and then at the end of the day we can spend some time. packing and implementing what we've learned and trying out what we've learned.

Matt McCormick: so we have a short break.

Matt McCormick: feel free to discuss or just take a look. and we'll continue

Matt McCormick: and in 10 minutes with the Giotto. tutorial

Matt McCormick: Did you want to? Do that in the afternoon today? under

Matt McCormick: Great. Yes, please. That's wonderful. Great idea.

Matt McCormick: Let's plan for us then.

Matt McCormick: around

Matt McCormick: 1:30 to 2:00 and

Matt McCormick: great.

Matt McCormick: That would expect more people could come we're not. here personally, Tom listen to us on the weekend or

00:10:00

Matt McCormick: on decided to the website, a plan just and if you have slides or if you just want to walk through. Maybe do a demo and slicer that would be great, too.

Bradley Lowekamp (NIH/NIAID Contractor): Is there any place where presentation's or documents or other files can be exchanged or saved from the presentation's?

Matt McCormick: the hackathon

00:15:00

Matt McCormick: So we have Igor. I see you join two and Ruben is here.

Matt McCormick: So maybe Igor and Reuben can introduce yourself.

Ruben Dries: All right. I'm Ruben. I'm an assistant professor at Wilson University in the Departments of computational biomedicine and television ecology. We are developers of shelter which isn't data analysis software specifically designed for spatial homework data analysis. And happy to be here.

Matt McCormick: Thanks for joining Ruben.

Matt McCormick: Igor

Matt McCormick: great. Thanks for joining us Igor.

Matt McCormick: And Ruben, did you want to present geodor or is George? coming

00:20:00

Matt McCormick: Okay, that's great. Yeah. Thank you.

Jiaji Chen: Sorry, I'm Ruben's PhD student and I am a lead developer on Charter Suite. So just pull up the actual presentation. Sorry. I'm late. It's

Jiaji Chen: sure screen

Jiaji Chen: this

Jiaji Chen: okay.

Jiaji Chen: All right. So John's Suite is a modular Suite of packages that we develop at the drieslab and they provide tools to process analyze and visualize spatial multiomics data.

Jiaji Chen: So this is usually going to be data that is generated from slices of tissue that are carved from blocks. These blocks are basically tissue surrounded by some kind of structural medium usually like paraffin or a viscous liquid that freeze as well and depending on the technology used the outputs can be pretty different but they are commonly a set of segmentations and raw feature points or they could be intensity images. And when you have these kinds of outputs, you can use these segmentations in order to aggregate your points and intensities in order to get a set of observations by observations by variables or cells by features. So it Matrix and commonly you will also in those cases get something.

Jiaji Chen: With the cells and also by X Y and Z to get a spatial mapping for your expression Matrix. So you can always convert from this over to this kind of format and depending on the technology you use you can have only this output. So this example over here is from a technology called visgen. With their scow platform where they have A single cell resolution data, they actually have I think. To our nanometer resolution information where they can actually resolve each of the individual transcripts. Those are represented by us as points data. And they also provide a set of segmentation information. So it's very highly resolved information. And then on the other hand if you're going for this kind of information you have 10x visium, for example where you have a regularly spaced array of dots

Jiaji Chen: that is able to capture transcriptomic information from the tissue that you've placed onto the slide. And then under each of these it's not necessarily single cell resolution information. You can have a mix of different cells that you're able to with further computation deep involve into different cell types. But the main thing is just that these are all spatial data no matter like the spatial resolution and there are spatial computations and also expression information level computations. They conform Downstream and that's what jotto specializes in it's built to in just any of the above types especially map data. And further analyze them. We can do things like cell clustering and typing differentially expressed Gene analysis and spatial expression pattern detection.

Jiaji Chen: among others So why are we doing forms in Shadow? Spatial transformations of the data itself will become more and more important in the near future. and that's mainly due to the fact that performing spatial analysis across any two sections of tissue from the same block will require that the data especially aligned to a common coordinate space. Even small differences that you make during the sectioning process from how you cut through that tissue to how long for example if it's an ffpe sectioning you float that on And a water bath it can result in even neighboring sections that are directly adjacent originally. They might be distorted when compared side by side. So these differences can make it kind of difficult to assemble multi-slice and if you're doing more than one different assay across multiple platforms. It can be hard to put those things into a cohesive 3D Volume for analysis.

00:25:00

Jiaji Chen: And the solution for this is to perform registration. Across either the data set images or expression information or maybe even a mix thereof. And depending on your registration results. You can have both the raster images. And Vector feature and poly confirmation aligned into a continuous hole. So ideally this kind of registration is going to be a free deformation. Based on sets of control points or you could also have outputs such as a deformation Matrix. But yeah find transforms are already going to be a decent approximation. But anyways, this is an example for example from STL line where they actually do free information kind of warping. Or you can see one data set two. They have this bulge out over here for both data sets in order to get some mapped together properly. another case

Jiaji Chen: if you use a simple affine transform or deformation or whatever that you apply it has to work the same way across the raster and the vector information in order to make sure that you can apply to all of your information at the same time.

Jiaji Chen: So for example, if you end up with something like a pickle file that is able to manipulate your image into the correct locations through a warping transform. I can also be kind of difficult to apply to the vector information as well.

Matt McCormick: I have a question for you George or a clarification. Maybe you can help me after the image data you're storing it in a table with the XYZ coordinates.

Jiaji Chen: So Jato provides spatial classes and methods for easy manipulation of data with 2dfine transformations. And these functionalities are all available from the dato class module package.

Matt McCormick: and the value Is that how it's always stored or?

Jiaji Chen: we offer

Matt McCormick: Is it also stored in kind of more of a? implicit way too

Jiaji Chen: yes.

Matt McCormick: Yes.

Jiaji Chen: over

Jiaji Chen: Or image data, So that I'll talk about that a little bit later. But I don't think I have any particular figure for it. But the main thing is that jotto deals with images and it holds a pointer to the file location on disk. and it does it through Tara which is the r package that it's on our package in the geospatial field that deals with raster images a lot. And we basically use Terra…

Matt McCormick: But that's mostly the spatial.

Jiaji Chen: which uses gdal in order to parse images that are living on disk and…

Jiaji Chen: a sample out regularly.

Matt McCormick: or the satellite type Formats, yeah.

Jiaji Chen: values from those images on disk So this is going to work for really any image format that you doubt does support which is good and bad at the same time. if there's a

Jiaji Chen: Yeah, mostly those formats. They do open a lot of Tiff images and also PNG. So most things that you would expect except if it's going to be more specialized formats than I think the compatibility is often going to be very spotty. but what we do is we take the set of every time we actually make a plot we Decide on a proper cropping and…

Matt McCormick: And what is a plot FLV for those myself Aren't as familiar.

Jiaji Chen: sampling method in order to get the fastest way to pull that information into a plot. fov

Jiaji Chen: So most of that data never lives inside memory, it's just pulled in whenever it's needed.

Jiaji Chen: Plot fov is just like for any particular. Space that you want to plot you can always set. Which particular areas you want to plot and this is usually in jotto at least it's done by a crop first on the entire object. And that defines which particular area you actually care about where your particular plot that you want to make. so that defines a geographical or spatial extent rather. do I have no, but it's a spatial extent that defines X-Men Max y Min and then you have the original image as well, right, which is

00:30:00

Jiaji Chen: Living on desk as just a set of pixel dimensions that you can take a look at but those are red into giato at the time of creation and there's a set of metadata inside of the Shadow objects. That basically reads in the pixel dimensions and immediately puts it into the coordinate space in some place. And then this is basically how you map an image into the coordinate space.

Jiaji Chen: on hand maybe a bit of coding something

Jiaji Chen: Yeah, so you can have an image itself. But this is a Shadow large image last though I have on hand and this is pointing to a file that lives on disk. So when Okay. Yeah, sorry.

Jiaji Chen: Yeah, we have a set of special transforms. So XY shiftings rotations rescaling to increase X and Y flipping and transpose sheer and affine. And along with that we also have some utilities which are basically just finding the bounding box of spatial extent. And for cropping out a section of that information plotting of course for pretty much all these spatial objects that I mentioned down here. Including one of which was actually just the total object itself, which may have a little less compatible with some of these but it can actually be F on transformed. So it's fully compatible with these more or less. So I just load in any subset of division data set. And I set the active spatial unit to aggregate.

Jiaji Chen: And I'm just gonna pull out the Java polygon information and also the image information just for the following parts. So this is me setting up a color scheme and also a set of line widths. And this is the show the individual transforms that we can do. So without Transformations. This is Shifting over to the right by a thousand. spin by 45 rescaling X and Y This is a vertical flip across the ZX or across the x-axis.

Jiaji Chen: And then transpose here. So we also provide a utility affine 2D class that you can make from any affine Matrix. So here I'm just initializing it with an identity Matrix. I'm setting in the extent the spatial bounding box of the general polygon information into the anchor which basically defines the starting Commission. And then after that, I apply all of those Transformations the simple ones that I mentioned earlier in a row to that I find object. So now it's encoding all of these and it's able to apply all these in a single application and a single function call.

Jiaji Chen: This is just showing what the offline Studio looks like. It also does a decomposition of to create individual transforms you can use if you want to. that's in the order of you the transforms that you would do with them and

Jiaji Chen: And this is applying that single affine transform across the Java polygon. And it doesn't miss a little step. So this is back to the images and this is pretty much what I've mentioned earlier where it uses Shadow large images. That's the orange class. It's based on the terrorist fat rasters. They're not loaded into memory until the time of plotting and Spat shift and rescale or XY shifting and scaling are supported by Terrace bat wrestlers by default and we inherit those functionalities but spin flip transpose shear and affine or operations will actually we handle those differently and we convert them into a shuttle affine image, which is the same except it contains this a set of affine transform metadata information. and then once that changes so that once you pull those values together.

00:35:00

Jiaji Chen: into memory, it actually also post-process it through magic,…

Matt McCormick: Judge a quick request.

Jiaji Chen: which is another image processing package in r with an image distort using affine And it also has some alternative extent and…

Matt McCormick: Could you increase the font size for us? Please thank you. Magic, is it like the image magic?

Jiaji Chen: crop methods so that these operations will respect both the expected post affine space and…

Matt McCormick: It's based on image magic to

Jiaji Chen: the UN transformed Source image.

Matt McCormick: Okay.

Jiaji Chen: sorry.

Jiaji Chen: Yeah, they are package that has bindings to that. Yeah.

Jiaji Chen: And then this is just me applying that affine transform to the entire jotto object as a whole. Where you can see that the entire object including some of the other information in a side of it. So the polygons the images you can barely see in the background and then just some preview of a couple of different genes. So these are the feature points that are also been transformed along with everything else.

Jiaji Chen: So currently the general image objects are not fully compatible with omitif because Most of the compatibility is kind of Relies on the genial drivers which I think that works just by checking across a bunch of different geospatial drivers for anything that will open the image. So gtf actually does work for some of the omitif images, but it does fail in some cases.

Jiaji Chen: And that's more or less all the stuff that we do with spatial transfer Moon Transformations and drato. if there's any interest in Learning more about Shadow or contributing to the project.

Matt McCormick: Thank you so much George.

Jiaji Chen: We are also hosting a free Workshop soon on August 5th to 7th And it's gonna be online free events you can.

Matt McCormick: This is very helpful.

Matt McCormick: Would you mind putting these in the hackathon?

Jiaji Chen: Sign up for using the QR code here and…

Jiaji Chen: our website is shadowsweet.com for more information and also tutorials on what we do.

Matt McCormick: And learn from it this tutorial here.

Matt McCormick: Thank and if we wanted to run that run this tutorial How would We get set up installing and…

Jiaji Chen: Yeah, thank you for the chest present.

Matt McCormick: running it. Maybe you can get.

Matt McCormick: okay. cool

Matt McCormick: Yeah, thank…

Jiaji Chen: Yeah, yes. Okay. Thank you.

Matt McCormick: Any other thoughts on I know you explored.

Matt McCormick: integrating czar omizar

Matt McCormick: into Giotto are…

Jiaji Chen: Yeah. I'll also include a set of installation auction.

Matt McCormick: what's your take on the current state of the situation there

Jiaji Chen: Thank you.

Jiaji Chen: we wish that there were more Tsar compatibility with our directly right now. I think that is a little bit difficult. There are some projects that are trying to get native compatibility with our but I think they're having difficulties with hierarchical. files

Jiaji Chen: and I think there's also some issues right now just with the difference between 36 32 bit versus…

Matt McCormick: Yeah.

Jiaji Chen: what are or 64-bit, right?

Jiaji Chen: Those are the main things right now and…

Matt McCormick: Right, right. and when you're taking for the Imaging data,…

Jiaji Chen: the way that we use images right now is through Terra. energy

Matt McCormick: especially Is there essentially Metadata that comes with it when you load it through gdal or…

Jiaji Chen: Yeah, so I'm not sure how easy it is to get gdal on board with also the changes in the bioimaging field.

Matt McCormick: imagemagic. gutera

Jiaji Chen: If it's an omitif. There So for spatial images I think there is some metadata that does come through sometimes especially if it's like to do with the spatial extensive information. So if you do save it as a geotif, for example, it will have a set of spatial mapping coordinates that the image automatically goes to whenever you load it with Tara.

Bradley Lowekamp (NIH/NIAID Contractor): So I have a question just I'm not too familiar with some of these are our packages being used but how much of the image processing analysis and…

00:40:00

Jiaji Chen: As far as other metadata information,…

Bradley Lowekamp (NIH/NIAID Contractor): interface with gdal is done.

Jiaji Chen: I don't think it's carried over.

Jiaji Chen: We basically have to add those By Hand by parsing through the xmls from me tip for omizar.

Bradley Lowekamp (NIH/NIAID Contractor): Out of cool adecore or…

Bradley Lowekamp (NIH/NIAID Contractor): you loading up the whole image to process usually loading does gdal for five functionalities. You can just get the resolution or the size or the chunks that you need, right? And is it parallelized with your framework?

Jiaji Chen: Yeah. I wish I knew a little bit more about that, but I think most of that is done entirely on desk and it's done in chunks at least by Tara and it is performed out of core. So you're able to work with the entire images. We are able to basically take these polygon segmentations and aggregate intensity images across the entire image without having to load all that in at the same time at any one point. But all this is mostly handled with Terra. It's only the parts that have to do with specifically plotting information when we pull in sampled. subsets of the entire image basically so we just use that as visualization, but if it's the deal with Actual image processing that kind of stuff is handled by Tarot.

John Bogovic: Because I have another quick question about. How have you struggled if at all about interoperating between different softwares with respect to how Transformations are stored has that been pretty straightforward or…

Jiaji Chen: Trump

John Bogovic: have you not had to deal with that much? I guess yeah, do you have any wants or desires when it comes to storing Transformations on disk?

John Bogovic: Yeah, I guess that finds I mean really I mean anything but yeah, affine's let's say concretely. I think you said something like you will store. Affine metadata along with the affine transformed image, right?

Jiaji Chen: Transformations on disk could you define more like what you mean by Transformations like affines or

Jiaji Chen: yeah.

Jiaji Chen: Yeah, I'm not sure if this is a convention already but there's the X and Y shifting that needs to happen right with affine transforms. I'm not sure if they're supposed to happen before or after the initial affine transform up like the two by two So getting that order in the right order has sometimes been confusing. There's also situations where

Jiaji Chen: For example, 10x one of them. Most popular platforms for generating this kind of information. They pretty much only provide for xenium their subseller at workflow. There's some cellular platform. They provide their H&E images. A separate from the rest of the data side because you basically have to do it after the zenium run and that means that you have to do an affine registration by hand essentially. So they provide an affine Matrix, but it's not entirely clear how to use it and it's a little bit confusing sometimes just because they have both this affine transform and…

Matt McCormick: Do you ever save out the Transformations that you're using?

Jiaji Chen: a separate Micron transformation that you have to apply at different stages of that transform.

Matt McCormick: Serialize them. We're just to find them in code.

Jiaji Chen: Yeah more information there about which order things should be done and would be helpful as well.

Matt McCormick: So in your tutorial you had your applying affine transformations. do you ever serialize them out to disk. And then load them again. Later.

Jiaji Chen: Do I ever sorry? Could you repeat the question?

Jiaji Chen: Yes. just

Jiaji Chen: bring that up again.

Jiaji Chen: So yeah, the affine transform itself is saved as metadata to The actual image it's sorry.

Jiaji Chen: Yeah, I find transform is actually saved as part of the job image whenever it's added. so you basically have that I find matrix in here you apply it to the Joel image and then the Java affine image and then it saves it into a slot and when you save that it's based all of these General images or affine images they're just a pointer to a particular file on disk and also The metadata information that's saved along with it including the affine transform, so it's not actually changing. The source file at all. It's only at the time of plotting or whenever you want to show a visualization of it that it is transformed. It's not fully implemented yet,…

00:45:00

Matt McCormick: Great, great and…

Jiaji Chen: but for cases…

Matt McCormick: what you can save your objects discs,…

Jiaji Chen: where you want to work with the raw data itself,…

Matt McCormick: all right.

Jiaji Chen: we are intending to make The polygons basically through the inverse a fine.

Matt McCormick: What is the format for that?

Jiaji Chen: so that you're able to apply directly on the original image without having to Do any destructive transformations to it?

Jiaji Chen: Yes.

Jiaji Chen: So we save it as a couple of different files right now. It's both an RDS,…

Matt McCormick: Is there any way to read those files from other languages or?

Jiaji Chen: which is the defaults are serialization format. and also a couple of geospatial formats SHP files

Matt McCormick: other question

Jiaji Chen: So they should be pretty standard across the geospatial field.

Matt McCormick: I want to take all the time here.

Jiaji Chen: But I haven't tried in other languages.

Matt McCormick: Ask another question then. You have the affine we talked about the affines. Maybe you touched it briefly here, but what other types of transformation the deformable ones What are the different parameters that you support now? And you'd like to have in the future?

Jiaji Chen: We don't support any others at the moment, but it would be nice to be able to have. things such as this where you're basically mapping that image to a set of I guess splines. That you are able to deform based on control points or something. I haven't looked into it fully yet, but I think the magic package. has

Jiaji Chen: Has a way to basically control points for to Target XY The Source,…

Matt McCormick: cool John question for you Now that I can start there are many different Transformations that could be defined by.

Jiaji Chen: in a series like that. And then perhaps it's also able to do a free deformation the way that we do the affine right now is just we set up three points at the corners. And then it's just like we take those three points Source Target and…

Matt McCormick: a set of corresponding control points it could be used to define many different like George was saying that fine or…

Jiaji Chen: then it's the deformation directly from there.

Matt McCormick: some deformable spline base polyfine what is currently in the transformation proposal for these types of transformations

00:50:00

Matt McCormick: That's good idea.

Matt McCormick: Hey.

Matt McCormick: What's up? a few new people have joined us.

Matt McCormick: You want to introduce yourself? Anastasia

Matt McCormick: I didn't scare Anastasia way. you introduced yourself. Would you like to introduce yourself to the group?

Matt McCormick: Thank you so much for Michelle.

Matt McCormick: Is there anyone else who joined recently? to introduce themselves

Matt McCormick: we is working on getting connected but are there any other questions or discussions with short for George?

Matt McCormick: and Giotto

Matt McCormick: great.

Matt McCormick: and then

Matt McCormick: Welcome Tao.

Matt McCormick: Is spend your mic? Might not be working. Or you're muted yourself permuted.

Ben Murray: Yeah, okay, Yes. So yeah, I had a different meeting link. So I think that's why. Knocking on the door knows letting me in…

Matt McCormick: But my apologies.

Ben Murray: but I'm here I realize we're already four minutes past the starting time. So shall I just jump straight in or did you want to do a little introduction first?

Matt McCormick: Yes, Go ahead and jump straight in.

Ben Murray: Super okay.

Ben Murray: Cool, so everyone can see that I presume. So thank you very much for hosting me for this talk. Please forgive the lack of an interactive demo. I think we kind of put this all together the last minute and so I can at least present everything to you. And the goal here really is to give you a holistic discussion over my pre-processing technology where it is now and where we would like it to go.

00:55:00

Ben Murray: So who am I my name is Ben Murray. I'm a research follow at KCl. I do deep learning across various different image modalities CT MRI Dermatology photos various things. I've been a call member of monothe on the start. I was heavily focused in the design philosophy of monai and the design philosophy if it's a term that you haven't heard before is focusing on

Ben Murray: why the design should be the way it is and how we reach consensus on what the design should be based on for example principles of existing libraries and what we think a good design decisions and poor design decisions that other labels have taken also I've worked on a number of key features in

one eye and a member of the number of working groups primarily relevant to today. I designed the lazy resampling feature, which I understand people are interested in hearing about and I'm also leading the feature design for the geometry support in the preprocess. and motivating statements for this deep learning preprocessing at least that involving images or geometry data.

Ben Murray: Really is a kind of Graphics Pipeline and I feel like the field should learn the lessons from Decades of best practice, but quite a few of those lessons haven't really been taken up. So we find that most libraries. Don't have the notion that you can compose lots of features together, or if they do it's added in a relatively.

Ben Murray: Limited way as opposed to a completely general purpose way and transforms aren't built fundamentally. To be composable in operation, even though you can put them all together in a pipeline and they run one after another left. Also, probably we should be thinking about learning from geometry presentation from best practice in various modalities and

Ben Murray: yes, I think that if we look at our Graphics pipelines work, there's a lot of useful information as to how one would design a preprocessing library for deep learning

Ben Murray: So just very quickly for those of you who aren't familiar with the graphics pipeline in computer graphics and the notion of composition. So computer Graphics works with homogeneous coordinate systems and homogeneous matrices in this is a very clever representation of multiple Dimensions typically two Dimensions or three dimensions in the context of what talking about that allows you to describe a whole different slew of transforms in a very unified way and applied them together and Get composed and efficient ways of transforming things. So for example here we start off at the bottom left hand corner with an identity Matrix.

Ben Murray: And the first thing we're going to do is we're going to apply a scaling operation to that. So this is the you sing my mouse.

Ben Murray: That again the scaling operation here and then we apply a rotation operation here. And then finally, we apply a translation which moved this image five picks the five units along and five units up and what we can do is we can take these two transforms. I mean when you compose an identity Matrix with a transform you get the transform, but then we can take subsequent transforms like this rotation Matrix and compose them together.

Matt McCormick: The software…

Ben Murray: So we get these scale followed by the rotation and…

Matt McCormick: then we're not seeing your mouse.

Ben Murray: then we can also composing for example this translation Matrix.

Matt McCormick: Now I saw that green one. Yeah.

Ben Murray: So now we've got a matrix that represents the scaling operation and the rotation operation and the translation operation in that order because these things are order specific.

Ben Murray: Here you see it. I can draw on it. Yeah, I'll just drawer attention literally to the things which until so yes.

01:00:00

Ben Murray: Adult element. So one of the

Ben Murray: Yeah, okay. yes, of course. That was why. So I was always drawing. Yeah, I was trying to I can do it. Either way. I can wave my mice around my presentation I can draw us. So these homogeneous matrices depending on the operations that are being composed into them. Typically, they are invertible for most of the operations that we would want to perform on image data when we're doing generalization such as rotations and scaling things like that. So we can also. Starting off with the compose Matrix from the previous slide We can invert the operations that we applied in the first place. And if we take all of these that we take the operation with everything applied and we take all of these inverted operations and we apply them in the opposite order will get back to where we started.

Ben Murray: So it is everyone basically familiar with that. Do I need to go into any more detail about that? All basically?

Ben Murray: Other single basically we can always come back to so.

Ben Murray: nothing exists in a vacuum. There are lots of different preprocessing libraries out there aside from monizone and monizone obviously certainly was not the first what you tend to see depends on the feature sets across the different libraries. So for example libraries that focus on raster data, it is generally the case that each transform tends to be Standalone. This means that if you perform a lossy resampling by that transform that loss of information accumulates over the stages and though there will be operations often optimize for transform type. So for example, if you can do things like flip operations or rotate 90 operations as a tense or operate operation, then you'll do it that way because it's more efficient to be it's not lossy.

Ben Murray: And some will require interpolate operations and somewhere require full resamples and even such as defamations, which I understand is a field that you guys are very interested in. Are done as general grid resamples, although you may have other methods for doing them.

Ben Murray: So some exceptions in raster libraries most of them will tend to have A Fine operation that allows you to I would like to scale by this much and I would like to zoom by this much and I would like to rotate by this much. The limitation of those is you can't interleave anything in between them in the different operations you're performing and also you tend to be locked into performing them in a certain order alternatively. You can also just Define a fine transform and run execute this operation with your own composed a fine transform, but then you have to have that capability to do that and deep learning practitioners. Do not always have the sort of computer graphic background.

Ben Murray: to be aware of these kind of things as rising has a stacked a fine transform, which is a wrapper that if you put other transforms in that are all its version of a fine transforms. It will do a similar thing to what Mona doesn't lazy resampling which I will show you a bit. On the geometric data side. There's a few different libraries.

Ben Murray: They tend to be primarily focused on Geometry rather than a combination of geometry and raster data. Some of them are really focused on graph formats, and some of them are just focused more on point clouds and measures. So these things to exist but what we want to do with monai is we want to unify all of the four beneficial features that various different libraries focus on so we want to be able to have differentiable transforms. We want to be able to have invertible transforms. We want to be able to

have composable transformed and we want to be able to have transforms that handle raster and geometric data in a

Ben Murray: a unified way and we want the monetary processing library to basically have all of these four features unified in a coherent way. So you're not struggling with different aspects of your data ingestion running on completely differentiable transforms, although you are obviously free to use whichever pipelines. You want my transform to their view. If you want them, please feel free to not using when we talk about differentiable transforms, we're really talking about using tensors in the transforms because then you can basically have the automatic tensor goodness of the filming performing.

01:05:00

Ben Murray: Autograd and being and having the preprocessing be part of the overall differentiation that occurs when you do about the past and it may be that there are other areas of differentiable transforms that are not covered by this. It's not my area of expertise if that's true for any of you here. Then I will also invite you to reach out to us. If there are features that you want to see that go beyond tensor preprocessing libraries being able to use tenses into

Ben Murray: So that's all I'll sound differential transforms. I'll say a bit more on invertible transforms because we have made all the spatial operations in monai invertible. And that's all the affine transforms a bunch of non-fine transforms and also cropping padding transforms as well and some of the defamation fields we have So there's lots of benefits of invertible transforms.

Ben Murray: what they really provide for you is that you can choose to map whatever outputs your pipeline Network produce back to the original data and that allows you for example, if you're doing a multi-patch sampling pipeline to see precisely how your patch is look with respect to the original images without you having to idle.

Ben Murray: I'll show an example of that in a little bit at least the results but really this is very much connected with this idea of composable transforms in Michael lazy resampling and it comes back to this motivation that pipelines generally treat spatial transforms discreetly. resample do the third thing example, some of these are lossless. So it's not a problem. It's just performance. Some of them are lossy. So you tend to get

Ben Murray: Noise, which is biased to the kind of operation you're doing added in it's not necessarily useful noise from a generalization perspective. And worse a lot of these operations are destructive if you're operating on raster data because you literally click data that has now fallen out of your new volume bounds or image bounds out of the data set. And then if you happen to perform another rotate or some other operation that brings that in there's nothing there even though from a bounce perspective there should be validating. so

Ben Murray: what we do is where it is possible compose these transforms and apply them. as I showed earlier in the composed pipeline we

Ben Murray: We composed together all the different a fine matrices and we perform the resulting operation in a single. so traditional preprocessing this is what you get for for example, if you don't have lazy turned on and my uses these meta tensors, which document All of the operations that have been performed on them and it's also part of the feature that allows monitor to invert transforms and literally each operation as it occurs is added to a list of Applied operations.

Ben Murray: With lazy preprocessing it's slightly different. So what we do is we'll apply an operation and the red operations here are operations that are being performed lazily and if it is a lazy operation, we will add the resulting transform to a pending list and we will compose all of these operations on the pending list so you can see here we perform. And the spacing operation then we perform the orientation operation the Rand spatial crop operation, but they're all just being added to this pending list and after each operation. We check various criteria that indicate to us whether we need to apply all these pending operations.

Ben Murray: And so as we go through this list, we reach a point where we've got all these pending operations and the next operation we have is random gaussian noise that is not a lazy operation. So at that point before we start the round gaussian noise operation, we must apply. all of the pending transforms to our tensities of the tenses here and that will take ABCDE and F transforms composable together perform the resampling and that point Everything in the pending list is reset and we are now in this sort of new reference space where these operations have been performed.

01:10:00

Ben Murray: So, why do we want this? I've made it a case for a couple of points already. For example on the top row here. You can see that we have performed a series of spatial transforms including zooms and rotates on this brain. And if you look at the traditional pipeline, you see that what we've done even though there should be brain available in this part of the image that brain was wiped out by the zoom operation. And so all we can do is perform whatever padding operation with Dean sit at the time. But more motivationally potentially is these resampling operations can be extremely destructive on labels because labels are generally resampled in the nearest neighbor way.

Ben Murray: And the difference between for example performing a zoom and a random rotate. Followed by inverting that zoom and inverting that random rotate on labels. So here we see a tumor and this is the information Ideally there should be no information loss. Ideally these should be identical but due to the fact that we still are performing some kind of resampling they can't be but you can see there is far pronounced information loss and artifacts and regions which have were connected and now unconnected regions that were apparent and that have now disappeared. Whereas if we do this in a lazy way we get far less destructive noise added with the round trip. So this allows you to compare things much more readily with your Baseline.

Ben Murray: Brown truth so I'll move on now to Geometry support. This is a feature that we are putting our

Ben Murray: First iteration of together either for this upcoming release or for the next release when we're talking about geometry here. We're talking about all of the data types that are purely geometric and I'm not raster-based data. So Point polygons matches graph data structures, we intend to build a gradual feature set for these. So we provide the critical things that people need in their pipelines now and then subsequently based on stakeholders and the user base reaching out to us giving us issues becoming Parts working groups and so on that will inform Us in terms of how we expand this state at this feature set and I mean they'll also with these different types of data structure. There's also different levels of operation that will be staged over releases. So we will do polygons and static

Ben Murray: is early but remeshing operations that occur for example, if you're doing defamation Fields will be done subsequently. So our design for this API is to make it seamless with the raster API. You won't have to say rotate raster and rotate geometry. You'll just say rotate and it will do the right thing based on

the data type. The API should have minimal changes. You shouldn't have to rethink the way you do stuff just because now you're loading geometry data and the hybrid pipelines where you have a mixture of raster data and geometry data should also just be simple. You shouldn't have to redesign your guidelines for that.

Ben Murray: We're also making sure that we provide pure geometry pipeline support So geometry will be a first class object. It's not some metadata that goes with a raster answer. It will be its own tensor types and it's not subordinate to raster data. Although there are some transforms that you use in a raster environment where the raster transform will inform the modifications to the geometry and dictionary transforms will all support mixture of raster and geometry elements.

01:15:00

Ben Murray: with defining some spaces in which these things are described. So we now have this raster space which is the space in which the sampling of raster data is performed and raster space operations will resample the data but they shouldn't change geometry. Although they might change the volume bounds of the data. We also have World space that's based in which the raster and geometry data operations live and all geometry data is in World space essentially and the spatial operations that we're providing for raster operations of things like rotate and Rescale which The actual raster data will have space and World space transplant components whereas geometry is World space only.

Ben Murray: And I wanted to have a nice diagram here, but I ran out of time to make one. So if you think about what a mixed pipeline would look like you would load your raster data, you would load your geometry data now depending on your geometry format, which may just be a simple Json file. There won't be anything that describes how your geometry data maps to arrest data in some cases. So we'll Ease of use functions which allow you to say? Hey, I've got this

Ben Murray: Sorry, was that someone asking a question? Okay, what was I say? Yes, so we'll have a ease of use functions that will help you make use of any metadata on your raster data. If you're loading from daikon daikon or Nifty or somewhere like that and map your geometry data simply to your raster data so that you've got your geometry data and your raster data all joined up in World space and after that the rest of your pipeline just performs these operations in this unified World space. And despite the fact that so raster operations have obviously this benefit to operating lazily geometry the case. there's no information loss by not composing.

Matt McCormick: Thank you so much. Ben. That was wonderful. Maybe a one question. I'd like to get in is what are you doing for serialization of the transformations?

Ben Murray: your operations, but because we have this ability to compose operations both will have the capability to operate lazy and that at least reduces resampling noise for the raster data. So to wrap up this is what we're building we want. We want a vision for preprocessing capability that brings these four powerful features together and we have a single coherent design for them which makes your use of them regardless of what feature set you're using as simple as possible. So we have transforms which are differentiable transforms which composable to avoid loss of rest to date in that raster information and these transforms will operate a unified way on the raster geometry.

Ben Murray: If any of this is interesting to you, if you think that there are things that you would like to contribute to monai in terms of proposals for any of these features that's discussed. There's lots of ways to do so you can raise issues you can raise PRS you can propose or…

Matt McCormick: Thank you.

Ben Murray: request to join working groups in…

Matt McCormick: The itk transform format is a strong Contender for export just…

Ben Murray: which case I would initially suggest you contact Stephen Elwood and here are some links from one eye and…

Matt McCormick: because that'll give you instant interoperability with ants elastics all the itk administration libraries and…

01:20:00

Ben Murray: thank you very much. I'll now take questions for anyone who would like to Ask any questions.

Matt McCormick: unigrad icon. And then they'll also with slicer.

Andras Lasso: What is the programming language that we implement this in and…

Matt McCormick: any thoughts on how difficult but

Ben Murray: So we will.

Andras Lasso: that it support for our execution on GPU out of core?

Ben Murray: pick I don't have the list to hand, but we are picking a Set of software that outputs geometry.

Andras Lasso: and for example for complex math, you need to go out to C++ or other libraries, I guess or there's a wrapping in Python that I guess you need to implement some algorithms. I don't.

Ben Murray: That people tend to make use in a clinical setting mostly at the moment and we are supporting those various formats that we have enumerated and we're also supporting very generic Json formats. If there are formats that people think we should be supporting we will happily take issues and PRS and add that to the design the feature set similar for export we will take the ones that we know we have proven need for first. And if there is a proven need or people are suggesting strongly that they have a whole user group that would love to have this feature then obviously that changes the way prioritize those but we will make it as painless as possible to load the various geometry data types that we support from the different formats that we're aware of.

Ben Murray: Yeah, yeah. I don't disagree with any of that.

Ben Murray: so this is a Python it's a monitor pie torch-based library.

Andras Lasso: Yeah, so we think it provide all these inverted you can conquertonate a number of warping transforms a few of them invertised and…

Ben Murray: And all of monai is implemented in pytorch. Although there are some who the kernels the transforms all support GPU execution.

Andras Lasso: apply a few wood in your ones and then you apply that to a volume to extract a singular slice. So you only evaluated…

Ben Murray: as well as CPU execution

Andras Lasso: where it needed. You can also reuse the same transforms multiple times the four bar transformingular transforms. You can have multiple pipelines and so on there's only if at what level this is reusable or not, but this is already a complete framework that works really well and…

Ben Murray: so For example,…

Andras Lasso: I think you haven't mentioned BTK and…

Ben Murray: yes, so python and…

Andras Lasso: in the beginning so I was not sure…

Ben Murray: pytorch is quite.

Andras Lasso: if you are already aware of it or…

Andras Lasso: not, but it's nothing interesting. And of course, I think he has where I just don't understand I think because I

Ben Murray: Specific in the operations that it does before that it does support.

Ben Murray: So for example, it's sort of in Congress that though. There is a resample operation. You must provide a grid as part of that resample operation, whereas it would be perfectly reasonable to have a resample operation that simply takes an affine Matrix. We do also accept contributions from people who want to provide

Ben Murray: very sort of powerful powerful back ends that we can wrap us around.

01:25:00

John Bogovic: And might you join the Julep the zulip at some point then.

Ben Murray: Might I joined the Watson. if you're talking about defamations in particularly defamations that get applied in the iterative fashion. And the lazy resampling feature of monai does not yet support composition of deformation representations, and it's something that I'd very much like to put in there and some motivating examples would be very good for putting that into proposed feature set for release.

Ben Murray:

Luca Marconato: Yes.

Luca Marconato: Okay, let's see if we can go full screen. Yes.

Luca Marconato: Thanks a lot for organizing this and invited me. I think it's a great opportunity to share what will own in our development and…

Ben Murray: Yes, I'm familiar.

Luca Marconato: also learn from you.

Ben Murray: I think in terms of…

Luca Marconato: Also. I really like that finally see in person.

Ben Murray: what we need to do to integrate things like that is we need people…

Luca Marconato: Beautiful but more in person some of the people that I met in gitaba…

Ben Murray: who are motivated to do it and…

Luca Marconato: where we just knew they username.

Ben Murray: who will be willing to sort of sit down work together As a suitable sort of it'll be a slightly design heavy process to make sure that it kind of meshes with everything that we already have…

Luca Marconato: So with this to start telling our story which is how within the framework, we explore the implementation that the carports and the stores the presentation and…

Ben Murray: but the more people…

Luca Marconato: they remember presentation and…

Ben Murray: who want to contribute and…

Luca Marconato: the work that I will show is not just mine is of our team.

Ben Murray: particularly if there are specific things that you really wanting to see then we do very much welcome your contribution.

Luca Marconato: So I want to thank Sudan Kevin Isaac about their team Josh and…

Ben Murray: I do have another meeting that started five minutes ago.

Luca Marconato: more in the last slide.

Ben Murray: So I'm afraid I'm going to have to dash but please reach out to me. I realize I have not Provided by email on the slide. So I'll just put it in the chat here.

Luca Marconato: So what we built is specially the which is a framework comprising of a series of python packages which provides a solution for working with special motionics Data Center, and we designed it so that it Bridges existing communities. So before framework is built across in particular free communities the s'verse community which developers and experience with single cell and Senator biological data.

Luca Marconato: Close to biological molecular data, then the ome construction which is experience with dealing with large images and standard formats. And then the Party core developers that experience with interactive visualization and annotation. So the framework we built is a company infrastructure for the stores manipulation and visualization and I want to Elite that is not a goal of the framework to be an analysis Library even if some of operations that we developed them that are common operation with

Special Olympics data can be used to perform analysis when the paper that we've wrote that we actually show someone analysis that are done entirely using our framework also want to Elite that we relying on existing Fighters the graphic pythons organic geographical information systems technologies as gidala and Igor's that are used them. Sorry Sid Albert Usain who's dropped that is used internally to leverage existing Technologies, so,

01:30:00

Luca Marconato: I want to give them a few slides on showing what is the framework about before Champion to the discussion with Transformations as to put everyone at the same level more or lesson. And so what we do in the specialty the framework is that we provide an abstraction by introducing some elements and these elements are raster geometries or a vectors geometries, and we realize that adding this composable elements and together also with annotations for this elements is enough to represent any Special Olympics data sets that is currently available and the annotations that we generate for us as a table are actually dense or sparse matrices tensors or data frames. What we do is that we brighten and read as close as possible to only in cff with the OEM is that implementation? This is not always possible reasons because geometries are not defined by the windy and specifications. So when this is not possible,

Luca Marconato: We extend the format and eventually we open that will be able to contribute to ncff with what we learn or on just wait until last summer parts are available and we can Implement them so that we are fully compliant. So in practice we write to that but also we write to parque for the points and the polygons in memory. We use standard data structures from python and data for a denotations, but also x-rays you pandas for antibiotics type of geometries and ask for having an infrastructure for loading a lazy computer.

Luca Marconato: And regarding coordinate Transformations, which is the topic of today. What we do is that we allow the definition of multiple coordinate systems. In this case. It just showed two but we can have this case where the pixel space with we Define the emails and the physical space which uses micrometers and we say we have a rotation that is defined the accordion Transformations. All of this is safe to disk a red from this according to the nzfa specification from and so on and what we do is that we also generalize the Transformations but for raster and that types, yeah, it's a show then there are and then in the end the usern has apis and the corresponding disc format to save this transformation and have an object that contains any number of special elements even zero even multiple of the same type with different current transformation with the one or more coordinate systems and an example that is shown in the paper.

01:35:00

Luca Marconato: that we could represent a spatial multiomics data center that contains 1bzuma without and succinum readouts with our Technologies from the next genomics with different properties and

Luca Marconato: And also we can store on this combining even annotations that can be drawn from the user are using and a party. Finally. The last slide on special data is that we don't just allow for presentation of the data but also we have some operations. So one is regarding how to transform the data. So is the topic of the next slicer but also we have functions to query the data with a bounding box or with a polygon or a multipolygon. We have an interface to deploying to pythons so that we can easily generate that data loader for the data and the trainer model and then we were super to transfer information between the different layers and the aggregation operations into Target geometryism. But

with this said that I would like to talk regarding Transformations, what we do in the library is that the couple they raw data storage and The annotation metadata, we do this stuff for a series of reasons for some because this is what ncff does.

Luca Marconato: Second is that in this way? We have a lightweight representation of a transformation that doesn't require to modify the data and then one can say upgrade the transformation without adding to tasks with data. So what we do when for instance we visualize the data using our plugin the party special data for instance that was used to create this visualization. What we do is that that's the metadata and we let the front end deal with this using the parties apis and while turning the next hour will and we'll talk more about this, but we can also transform the data and directly so you can call one of our apis that transform API that takes little data and now updates so that there is a no need for having a transformation metadata anymore more on this later. And then now if we plot this emails in this case, we would have also this padding in the corner so for our implemented preparation, so the code that we provide can be seen on the right side. So

Luca Marconato: I'm doing it from a special little set transformation and translation. We can define a translation transformation very conveniently and we can simply set it inside the element. What is done is that the transformation is placed inside the metadata element again more of this later when we want to call the transform operation. We can import this API from specially the packets and we can simply call the function saying that we want to transform to the coordinate system of a transformation that we Define before the target of the transformation and what is given is the Transformers. Otherwise they could have said I don't want to transform the emails. I want to just retrieve the transformation and then deal with it manually and this is also possible.

Luca Marconato: so a few points to notice before continuing the force is that the transformation year is not defined on czy X, but in this case we chose to Define it from to x y and in the code that we actually made a difference always we say that with translation is translating Y and said even if the emails who contain different axes, so this is one of the flexibilities of our memory apis that will discuss in details in the next slides.

01:40:00

Luca Marconato: So Visa is an example of this we can using our apis and easily create composer rearrange the transformations in the example here. We Define a translation a scale and a fine transformation in a sequence and here you can notice that we have the same terminology of omni and CFS specification. So in this case, we have a translation operating the a scale defined this time on the x axis and I find that goes from the XY to the y x axis and then a sequence that combines everything if we print them the sequence what we represent is Visa and if we can also as they say the rearrangements the Transformations so we can say I don't want to have a sequence. They want to commit this on a fine Transformations and they care about these input axes and these output axis. So this is possible and you get and a fine transformation as output and also a key point is that the memory Transformations are defined independently.

Luca Marconato: Axis of elements were applied to so this is the key difference with nzf Transformations that will show in one of the next slides. So what we have is that here I Define a map axis operation. That's just swapsa then X and Y axis then I Define a scale on the set axis. Then I Define a sequence that maps that combine some axis twice. So this is a put into an identity and then there is a scale on the set. Then I decide I want to materialize these sequence transformation on the axis that they care about and this

returns. These are fine Matrix on the access they care about so you can see that when I Define the sequence I didn't even know when I Define the map boxes that I ever said access when I Define the scale the note that I get X and still the results is what it would expect when I combine this sequence and the advantage of having this is that if we have a different data types with different axes like

Luca Marconato: In this case, we read the other store and we read our specialty objects from which we can access the various elements and one element is my emails. Another element is my points my images as see why my points are the X Y axis. So if I do now get transformation from the images and then I set the center transformation to the points with this system. It works because we Transformations as I said are defined without knowing which axis we will be operating to and this gives us some extreme flexibility and economics to the users reading and the right into this term still is not done in every format is delegated entirely to the ncf transformation. So if we write the data and we did again the transformation will be written as FF and red into our memory format, but we can also say let's replace the transformation without having to us without

Luca Marconato: We want to be able to do this lightweight operation. So this is possible and this also works for vectorate and for us the images and what is done internally is a bit more developed than this but to synthesize it is that we take our sequence Transformations that we Define before we call two nzff now we get a new class which is closer and cff Transformations with transform into dict and we dump it into a Json. So what we get out is respecting the direction of a specification that maybe now is a bit different because I wrote this code one year ago. So I haven't select if something was updated but this was the respecting and I think also now there's a draft specification it is out. So we have the find transformation and then we have a set of axes for the input coordinate system and the output coordinate system.

01:45:00

Luca Marconato: as you can see when we do the conversion between our memory format and then cff format. We need the excellent information in particular. we serve the input and output access because we actually need to know which is the input and output coordinate systems and to keep things simple for the user. We make a series of assumptions that can be overridden the forces with the fault coordinate system. If nothing is specified. It's called Global and then this design will see some changes and to make it more transparent and the second is that we Define at the fault that you need to call the unit the user can choose to put micrometers or a pixel or other elements. So the fourth implementation that we did them which is a public available is mirroring. So this was what they show now is what is implemented now, so now I make a step back. I want to show how we go to there. So the first implementation that we did.

Luca Marconato: Was not the one that I show you of course that we try to mirror which ncfs specification as close as possible and implemented to classes to series of classes one for coordinate systems. And what is related to coordinate system like axism and one for coordinate Transformations. So vision is the code.

Luca Marconato: Apartment of the code that shows how your operate with nzf coordinate systems. So we Define axis and coordinate systems and the accordion system object may look like this with a name and a series of axis that are named the typed and with a unit and then we have a series of coordinate Transformations. And when we did implementation, we didn't Implement all of them. So we skipped them up index because we have already map access and with the side, let's start from this and also we actually skipped inverse of invite action and we also skipped displacements and coordinates and probably

displacement coordinates are the most important ones. So we wanted to help them in the future for the rest is important that will be the further in term of priorities. And so I show the class and ZF identity. So it is from a base class and it has a series of metals one is from dictator to be able to parse it from the Nick that

Luca Marconato: From the season we have a two digit that is an user for a civilization and web initializer that in this case gets the input and output coordinate system, but not the other things because it's just an identity transformation what we did then is that one comment. So there are many improvements that could be done to be sent to this code one of these for instances to use pedantic as some related work from a Davis in identical me and cff for version 0.4 so that there is an automatic validation of the parameters that we do so we didn't do this in this case. What we did is that we didn't ask them after doing the IU partner. We implemented the fourth set of operations and such as a functional computer in person and then some functions for validations so that when we want to apply the transformation we

Luca Marconato: A sec that the data that we have is compatible with the input and output coordinate system and Transformations. We have to a fine function since we were skipping the newly Transformations, we could transform everything also by dimensions and map access to a fine and finally we're implemented them a transform points the functions to do some experiments and I took before that we separate now the current transformation classes and the Transformer functions. So now in the new classes this transform is not present is done with different API. And so when we implemented this but we add a series of drawbacks. So the first one our API is when we apply the practice where two verbose because the users to specify or important the coordinate system input output even for identity. This was to specified the second. Is that there's no see access cases to be spec.

01:50:00

Luca Marconato: Either and this created extra repositing if you have an email that you want to align with some geometries with created Soma some problems in terms of the positive second Transformations, as I mentioned before could not be moved around easily if we had the XY points and we want to put a transformation from points like a scale inside us see why Ximena this required a lot of code and I put a glimpse of the code here on the right side. So here we actually try the two ways to do this on the top we import the axis because they need to be passed with Transformations. Then we Define to affines want to go one from XY to Cy ax so easy, and then we also try them. let's see if there is an automatic way to make it work without having to define a fine Transformations or without adding to have a function that finds automotive and Transformations. So realize that going from XY to see y x now is a technical it's just to give an idea.

Luca Marconato: Of the challenges, we could combine a by-dimension with map axis for actually want to keep and a fine which gives us zero padding for the axis. We don't want to keep them and then on the way back we could use a map axis. So we experimented the bits and it worked but it was to the boss. But one of the reasons why we decide to switch back is that we found some complexities with the sequence Transformations that I will service lies later following the link. But basically when you define a sequence transformation at least back, then you could avoid to specify the input tax in output taxi for the sequence Transformations these two avoid their positive, which is very well still the word cases thing with Transformations that could be Amigos if the input and output access, we're not specified like a map Axis or on a fine because if you have on a fine that is a super free you don't really know if they

Luca Marconato: They is swapping some axes or not and that we could infer some of these missing information recursively but there were some keys that were uncovered. The code was becoming very complex. So we decided let's try a different approach. let's use as much as possible or the class that we have for input output especially also because the semantics of the values classes like mapbox is a fine. So on is that it's very well defined, but let's try to see if for our memory needs we can have different classes. So when is where we try and we were satisfied after to have a different classes in memory and that we implemented the series of simplification. So the first one is that we Define transformation independently of the input alcohol coordinate system, they will be applied to it. So this is what they show before we can transformation without knowing what the transformation will be applied to.

01:55:00

Luca Marconato: Example we can say the transformation is a translation on the excesses access by side and it interprets us if there is a next translated by five. We don't need to know if it is a y not. Implementation also can add them in the presentation. We did this. We don't use coordinate systems to Define Transformations. We can Define

Luca Marconato: Transformation for instance an identity without knowing the input/output coordinate systems, but in doing this we require extra arguments because we still need to have knowledge about the axis and examples is as above them. If we Define a translation we can say this operates on the x-axis we Define a fine with we need to know not just the input access but also the output axis. So this is required what it didn't say is that Transformations are still and dimensional and any order of access axis any type of access and any number can use them with one comment in special data. We don't use axis about the cxy and see that why accent so actually we have some validation that we use a validate against this but this is a constraint we could remove that didn't say in the second thing that I/O is still done at the ncff Transformations. Thanks to our conversion functions.

Luca Marconato: So we have conversion function between the immortal Transformations that are subclasses of Base transformation and what they should be for the based transformation.

Luca Marconato: So how do we actually Bend transform elements? Because now I show out to define the classes. So even now we can have some simplification. So the first is that right now we didn't Implement a new linear Transformations. So all the Transformations can be turned into a fine function this simplifies a lot of the code and this also can be relaxed and we want to actually include Benin transformations in a later stage and also implementations we decide to skip for the moment the buy Dimension and you have transformation and rotation because they could be used with the one could Define a fine transformation for this case. And also we already knew the input taxes to our elements because we have some elements schema. So we know that why X 3D images are always see why excess see that I skipped on Xiao.

Luca Marconato: Is y axon and so on to the points XY 3 points XYZ?

Luca Marconato: Finally and this is let's say they technological trick that makes everything work is that we can find the output xis. No in the input taxes and knowing the transformation calling a function that is a get current output axis. And this function takes the input taxes and gives the water valuable taxes independently. What's transformation is put if it's a simple one like an identity or is a complex one like a sequence of a fine.

Luca Marconato: so let's see now how we deal with missing axis? Because if I show you this function get current output axis, it may raise it some questions for instance. What if I ever translation between X and Y

and I say, I would like and a fine transformation because c that gives us output axis c z y x so we have an extra axis. That is the C present in the input and the output and Anastasia's axis. That is the said present only the output. And this is Delta in this way the C is passed through because it's present both as input and output axis, but is not defined in the transformation this add them is ignored because it is present only the output axis.

02:00:00

Luca Marconato: This is not always possible. If the user passes and a fine transformation for instance if the user wants to convert this translation in a fine transformation that contains also the C there is no way to deal with DC because it's not present.

Luca Marconato: in the original data or at least that's what we thought was a good idea for our use cases. So we throw an error input access must be a subset of the output axis. We saw that with this simplification we could make the method work a second case that doesn't work with validation is if there is a fine transformation was this is introducing a new access. So in this case, the find transformation takes the X and maps The X to X and why so now if you want to materialize the fine transformation between XY and XY we get the same error us above because why is not present us input. But even if we remove this for go to remove it we get an error why access is not in it's not an input of the financial information but it appears as an output. So basically we can't use the strategies about

Luca Marconato: Passing through the Axis or ignoring the axis because the axis is actually present as output transformation. So this leads to an ambiguity and we did some edit testing on this approach by composing the different type of Transformations. We put thinking into it and we think that it works and we didn't see any problem with our use cases combining so the 3D or any dimensional transformations. So now that we saw implemented our transform function. Let's see we actually form given an element where do we put the transformation and how do we transform it?

Luca Marconato: So our element is in this case an image contains some metadata and metadata is having a transformation that Maps the elements to Target Korean system, which the fourth name as said before is the Global Courier system and for us according system is just the Stringer that tells us what is the name of Korean system. So if we have a second element we can have a second transformation that maps in the same coordinate system and we could also Maple the same element to different coordinate systems. So whether this one directional way of defining transformation between elements to coordinate systems so this is the same also for Vector elements not just for images and as a set coordinate systems are just a string. We store the transformations in the elements metadata.

Luca Marconato: In this way so we can access an attribute of element and as a dictionary target coordinate system transformation and then we plan to store also some optional information for Korean system. So if we know the sample ones as the set of axes with a set of units we can choose we want Implement Visa so that we have a dictionary that says sample one has got this access that are notated but the system as it is implemented works also without this so this is how we look like we can look at the coordinate systems lot inside our object and we will have a dictionary containing the same name so they can be present. They can also be missing and then in that case there is an nzf coordinate system object to reads the information.

Luca Marconato: So one question before I continue format, I think that I have three minutes left to have the alpha an hour pasta. I have more slides and so, what's the best way to deal with disable 10 more slides with so many missions, so it's actually maybe foremost lies. So if you And also don't worry.

Luca Marconato: So in that case yeah, uterus got us representation. So yeah, we will have agreed that I can run a bit over. Okay. Thanks.

Luca Marconato:

Luca Marconato: now that they Define where we store the transformation within the object I can tell how we implementing the transform function briefly. So what we do is that for us their types. We use dust image and the interprep a fine Transformer and for Vector types. We use Gio pandas apis. So Japan does Juve series a fine Transformer. This can be optimized it as advantages. So the first one is lazy and some wise we can transform even use images but for instance the one could use GPU acceleration and the second is that is not lazy. The one from Japan does we could use Japan us and we want to do it and these are some examples on how it looks like so we can take a Transformations and transform in individual layers. We have in the special framework a series of plotting functions. So if this is made by our plotting functions, so in this case, you can see we are applying and find transformation with

02:05:00

Luca Marconato: Station and a sheer part to them raccoon and here there is a few rotation in various translation and So it's ergonomic words to get the results out of a Transformations and so limitation of the memory classes. first one

Matt McCormick: Or maybe speak for everybody look at us that was phenomenal.

Luca Marconato: For what implemented that we didn't find limitations,…

Matt McCormick: And I really love the ergonomics with special data and…

Luca Marconato: but we have to implement also other parts. So for instance then no linear Transformations are not implemented.

Matt McCormick: what you've done and your explanation. There was really good.

Luca Marconato: So this is one of the points that need to be addressed.

Luca Marconato: We have work in progress to be us and it's a master students in our lab and…

Matt McCormick: yeah, really wonderful and

Matt McCormick: I want to give a Witcher a chance to go…

Luca Marconato: Heidelberg and that is working on interfacing to external methods like XI and…

Matt McCormick: but I do have some really itching thoughts that come from that one is of course there's a lot of Parallels Lessons Learned with itk and…

Luca Marconato: eggplant and also in Auto Transformer the transformation that is then derive from this metal. So sometimes these Metals we apply directly there the transformation…

Matt McCormick: avoiding and…

Luca Marconato: but we are interesting actually extracting the displacement fields,…

Matt McCormick: going and avoiding pixel data that pixel Transformations and…

Luca Marconato: and we apply to developments.

Matt McCormick: also thoughts too with the and x-ray coordinates and how you build compatibility with that. And also, you saw Ben's presentation with manai. for example and kind of simulator similarities there where they had the for image data kind of this. I forget How he referred to it, but not the full and the fully functional Global space, but you have

Luca Marconato: and the same mentioned transform, let's use a limitations of a transform function. So the first one can be optimized with Adobe algorithms. So not just for find transformation, but for our assembling and so on and also GPU acceleration the second is that currently we only allow within the specialty framework czyx. We don't use even if you could have multiple images so there are work around but we decided within Anastasia the framework to limit to this access even if the Transformations are international and also we allow only specific order for foxes one of our elements so our labels will always be YX never X Y finally we don't really see as a special axis. So for instance embedding a single Channel limits into multisun lemons is not allowed by the transform function, even if it's allowed by the transform specification that we implemented.

Luca Marconato: Instead we prefer to use code like calling task array stock.

Matt McCormick: a DOT nrrd,…

Luca Marconato: Finally and…

Matt McCormick: and That mermel and…

Luca Marconato: is the last point there is the currently no breeds between nzf Transformations and…

Matt McCormick: I try and load them all then I…

Luca Marconato: x-ray coordinates.

Luca Marconato: And if you are familiar with x-rays, what they provide is that the image also coordinates specified for the pixels and…

Matt McCormick: Correctly interacting coordinate systems with all of them.

Matt McCormick: In this beautiful way, that's almost Magic. But it was actually a lot of work.

Luca Marconato: this implicitly is a way to coordinate Transformations.

Matt McCormick: Do you think? It's feasible…

Luca Marconato: You can store rotations,…

Matt McCormick: if this is specified well enough.

Luca Marconato: but you can store

Matt McCormick: Is it going to have all the information needed so that itk can just say?

Luca Marconato: You can store scales you can store translations you can do inversions.

Luca Marconato: So you have clipping the access to a way to store this information and currently we found a bit cumbersome that the memory we have an x-ray object and…

Matt McCormick: idk. read Of one of your exported transforms and…

Matt McCormick: it'll happily move normal files around just like it loaded native itka hdf5 transform.

Luca Marconato: and then we use nzf Transformations that are not gonomic. Okay, so

Matt McCormick: So go ahead.

Luca Marconato: yeah last four slides So I want to dig deeper into the limitation of X-ray and then I will drop up. So this is the current implementation of the coordinate Transformations. So let's imagine to have a fine transformation which is in this case with a Scala by K a translation by x y and rotation by Theta.

Matt McCormick: Wonderful. Thank you. Witcher did you want to Get started to have kind of took most of your time yet.

Wouter-Michiel Adrien Maria Vierdag: They're realistic there and other questions.

02:10:00

Luca Marconato: So let's imagine to have an image of this by 100 pixels and…

Matt McCormick: Maybe we can discuss today after the rest of the tutorials are done. Continue.

Luca Marconato: let's say the k equal to 4. So what we do currently is we take the image and we output as a Target image an email that is 400 by 400 pixels. We apply rotation adding corner paddings and we don't apply the translation because since we don't have a good Breeze between x-ray coordinates and transfer and the left Transformations, we always have to start from zero we can say the star from xui. that we recognize that is not a good idea for a number of reason for someone is that we are increasing the number of pixels and ssrily and also we start from zero we like to start from XY feel we have design. So now it's just matter to sit down implemented implementing it a way to add these output and…

Wouter-Michiel Adrien Maria Vierdag: All Can everyone hear me and see the screen?

Luca Marconato: output that contains still the corner padding. So this is still performing at a sampling for…

Wouter-Michiel Adrien Maria Vierdag: Yeah, okay. So hello,…

Luca Marconato: what is not

Wouter-Michiel Adrien Maria Vierdag: and thank you for the opportunity to present here.

Luca Marconato: So available to x-ray but only for a number of pixel that is at most the number of peaksolut images.

Wouter-Michiel Adrien Maria Vierdag: So yeah, I mean I also have included the whole majority team.

Luca Marconato: It's not like increasing tenfold.

Wouter-Michiel Adrien Maria Vierdag: So I don't mention all the names here immediately.

Luca Marconato: The coordinates will be using so the emails here will have X and…

Wouter-Michiel Adrien Maria Vierdag: But I mean this has been worked together with the owner party team and…

Luca Marconato: a 400 plus X. So it's containing both the scale part and…

Wouter-Michiel Adrien Maria Vierdag: also the spatial data team.

Luca Marconato: the translation part within the X-ray object and…

Wouter-Michiel Adrien Maria Vierdag: So yeah.

Luca Marconato: this is ergonomic and…

Luca Marconato: performance and

Wouter-Michiel Adrien Maria Vierdag: And so in short for the people that don't know yet or…

Luca Marconato: So the way we will do this is that as I said before we have the elements that have a transformations to Target Korean systems and…

Wouter-Michiel Adrien Maria Vierdag: are mostly working. In other ecosystems. For example, the art ecosystem and a party is an indimensional image viewer in Python.

Luca Marconato: still we don't support a certain cases in a special data.

Wouter-Michiel Adrien Maria Vierdag: We support interactive image visualization annotation and also analysis.

Luca Marconato: If you want to know what is the transformation between this special element and…

Wouter-Michiel Adrien Maria Vierdag: Although I have to say for the Anastasia's part,…

Luca Marconato: the coordinate system too or…

Wouter-Michiel Adrien Maria Vierdag: ultimately an apart is just an image viewer,…

Luca Marconato: between two special elements also coordinate systems,…

Wouter-Michiel Adrien Maria Vierdag: but it's very strongly connected to the python ecosystem.

Luca Marconato: you can find Mr. Summation. The graph transformation will be Traverse…

Wouter-Michiel Adrien Maria Vierdag: And with the way that the praise build up like you can easily do analysis and…

Luca Marconato: because thats and then Traverse and the mapping transformation in their Center transformation,…

Wouter-Michiel Adrien Maria Vierdag: then immediately visualize results.

Luca Marconato: but you can't…

Luca Marconato: then say this back to the store as a because importantly with the implementation we have now,…

Wouter-Michiel Adrien Maria Vierdag: So yeah at the…

Wouter-Michiel Adrien Maria Vierdag: of course, we take care of the viewing and…

Luca Marconato: we don't support this which is supported by NSF.

Wouter-Michiel Adrien Maria Vierdag: exploring of end-dimensional arrays,…

Luca Marconato: And the reason why we don't support data,…

Wouter-Michiel Adrien Maria Vierdag: but we can also lay the right data So currently support our labels points shape surface tracks and…

Luca Marconato: even if we want to be able to ever transformation that Maps directly the coordinate system together,…

Wouter-Michiel Adrien Maria Vierdag: vectors and…

Luca Marconato: so the coordinate system to look like Lisa that is exactly the reason…

Wouter-Michiel Adrien Maria Vierdag: there's also working progress for graphs. We do this all with very native python structures,…

Luca Marconato: why we

Luca Marconato: that they show before that we can't store a coordinates within the Amazon.

Wouter-Michiel Adrien Maria Vierdag: for example, a Lotus with numpy for example, and There is also of course x-ray white we can work with any kind of array- structure also sensor stores and so on but ultimately the conversion at the bottom level is numpy mostly endosk. And with all these components you can basically seemingly with exploration annotation and…

Luca Marconato: The solution is that we will drop the mapping between elements and…

Wouter-Michiel Adrien Maria Vierdag: computation together in image analysis and SF before it leverages the Scientific Python ecosystem. So everything that you see in the viewer and…

Luca Marconato: coordinate systems. We will only allow mapping between coordinate systems and…

Wouter-Michiel Adrien Maria Vierdag: interact within the viewer is available as apply for objects.

Luca Marconato: coordinate systems. So we with us allow the red arrow on the right. We will not allow the black arrows.

Wouter-Michiel Adrien Maria Vierdag: So regarding interactions in the Nepali fewer these basically need to events being emitted.

Luca Marconato: Neither. There are on the left. What we do is that we will assign like ncff does for its element according system,…

Wouter-Michiel Adrien Maria Vierdag: I show here a couple of the events for the individual layers.

Luca Marconato: but elements can also share codeine system.

Wouter-Michiel Adrien Maria Vierdag: And so for example there don't defense or…

Luca Marconato: So in the case of a blue and…

Wouter-Michiel Adrien Maria Vierdag: data lateral defense total Define whenever these changes there is a change you can basically connect these to a callback function that allows you to do anything.

Luca Marconato: the green images, they can be aligned by leaving the same coordinate system and the alignment is taking care of by the X-ray coordinates in the case of a purple image. This is not possible because the purple and…

Wouter-Michiel Adrien Maria Vierdag: For example here.

Luca Marconato: the green emojis are ever rotation.

Wouter-Michiel Adrien Maria Vierdag: I create a small callback function…

Luca Marconato: We can't be taking care of by x-ray coordinates.

Wouter-Michiel Adrien Maria Vierdag: where I basically check what is happening with the data on the layer?

Luca Marconato: So in this case, we have to Define according system for the image and…

Wouter-Michiel Adrien Maria Vierdag: I can see …

Luca Marconato: only for the image and…

Wouter-Michiel Adrien Maria Vierdag: what kind of index and…

Luca Marconato: then we can map this could be an assistant to the coordinate system.

Wouter-Michiel Adrien Maria Vierdag: label data is actually changing and what's being done to it.

Luca Marconato: Via rotation over a fine transformation.

Wouter-Michiel Adrien Maria Vierdag: So I'm first adding a point…

Luca Marconato: So with this sense,…

Wouter-Michiel Adrien Maria Vierdag: then changing it. You see the event being emitted both before and…

Luca Marconato: we will be able to address some limitations of representing any Korean transformation between any element and…

Wouter-Michiel Adrien Maria Vierdag: after the changes actually ccurred. And using that you can also basically connect it to anything regarding annotation and…

Luca Marconato: also addressing the performance and ergonomics to tasks from before.

Luca Marconato: I will skip this slide…

Wouter-Michiel Adrien Maria Vierdag: so on updating visualizations on the Fly.

Luca Marconato: because a bit technical but we can maybe discuss this.

Wouter-Michiel Adrien Maria Vierdag: And aparican also be easily extended using magic GUI.

Luca Marconato: So this is how we actually implement this and I want to jump to the conclusions.

Wouter-Michiel Adrien Maria Vierdag: We can create very simple widgets that allow you to run pretty much any function with a button.

Luca Marconato: So what implemented there is a series of ergonomics apis that address our Special Olympics use cases on this.

Wouter-Michiel Adrien Maria Vierdag: You can press the button and then it's Place change. So here if you change the x or…

Luca Marconato: We have ncf Transformations.

Wouter-Michiel Adrien Maria Vierdag: you create a different random image with a given amount of X with and…

Luca Marconato: And we also Define Transformations for points and shapes The Proposal we could discuss this during these days is that we could consider moving out of special data the code that I just described in a more General repository.

Wouter-Michiel Adrien Maria Vierdag: height. And also we can make use of QT widgets to make more complex interfaces as shown below. This is already an example of my prize space data and…

Luca Marconato: So can we use by other people and…

02:15:00

Wouter-Michiel Adrien Maria Vierdag: I will go a little bit more in detail later and…

Luca Marconato: also can be improved together and it can think of a series of tires tears to do Visa.

Wouter-Michiel Adrien Maria Vierdag: the fewer can be made to listen to events of security widgets and five Versa,…

Luca Marconato: So the force is that we could disaster at public outside the ncff business transformation classes.

Wouter-Michiel Adrien Maria Vierdag: which really makes possible and interactive workflow. We also have quite an extensive plugin ecosystem at the moment like 442 plugins.

Luca Marconato: So they read variety apis basically that are closer and we could add

Luca Marconato: think model and other optimization the second tier is that…

Wouter-Michiel Adrien Maria Vierdag: I have to say we are not responsible ourselves for maintaining all those plugins.

Luca Marconato: if you like our set of apis and with a couple

Wouter-Michiel Adrien Maria Vierdag: So yeah, you do have plugins also there that are not well maintained. But yeah, there's quite an extensive plugin ecosystem. Now coming to the price space data and the actual transformations in a party which is more the topic of today. And so first a brief overview of Nepali spatial data,…

Luca Marconato: With the definition of a transformation from a definition of the axis we could export both.

Wouter-Michiel Adrien Maria Vierdag: so the price spatial data was a plugin that we created for visualization of data within the special data ecosystem.

Luca Marconato: So then you can use the same apis to define a fine transformation combined and…

Wouter-Michiel Adrien Maria Vierdag: Main reason…

Luca Marconato: merge them and…

Wouter-Michiel Adrien Maria Vierdag: why we initially went with nepari is…

Luca Marconato: so on and the third Tierra it is a we didn't think of exporting out with transform function…

Wouter-Michiel Adrien Maria Vierdag: because all the elements that we have in the format are basically also available in aparri.

Luca Marconato: because in the end, we apply transform function to x-ray objects and to Japan's object.

Wouter-Michiel Adrien Maria Vierdag: So the labels Point shapes and…

Luca Marconato: So it's not some weirded attractors that one will ever to handle and…

Wouter-Michiel Adrien Maria Vierdag: I mean tables are not directly but with this event system, we can quite easily hook up like anything with n data to the actual viewer…

Luca Marconato: then maintain it's still something that is from the community and…

Wouter-Michiel Adrien Maria Vierdag: which then allows if I click your own it will update the fisherization for that.

Luca Marconato: in this way, we could address optimization points like using gpus or…

Wouter-Michiel Adrien Maria Vierdag: Here on the left part you see the coordinate system look already briefly talked about …

Luca Marconato: having a dog algorithms and we could imagine doing this to generalize the transform operation to be axis…

Wouter-Michiel Adrien Maria Vierdag: How is this represented in briefly quite extensively here.

Luca Marconato: because we just have said…

Wouter-Michiel Adrien Maria Vierdag: If you click on a particular coordinate systems this elements widget automatically updates and…

Luca Marconato: why X and also the final point is that independently on this I can see

Wouter-Michiel Adrien Maria Vierdag: shows you like what kind of elements are available.

Luca Marconato: Of having conversion functions between our Transformations and…

Wouter-Michiel Adrien Maria Vierdag: Also if I click here on the layer, we have tables that can annotate these individual space elements and…

Luca Marconato: it came out of deep Nepali also as was mentioned before the transformation used by the next genomics.

Wouter-Michiel Adrien Maria Vierdag: if you click on the layer you immediately see whether there's a table and…

Luca Marconato: So we have some code for this what is scattered around we could formalize it into our organizable place.

Wouter-Michiel Adrien Maria Vierdag: data table actually anything that element. And you can also switch in case you have multiple tables annotating the individual elements.

Luca Marconato: So with this I would like to conclude them. With some Shameless advertising if you're interested in…

Wouter-Michiel Adrien Maria Vierdag: There are also other parts just to briefly showcase a little bit with an annotation widget and…

Luca Marconato: what we do in the special data Library. We are actually having a conference with acid our first conference.

Wouter-Michiel Adrien Maria Vierdag: how that sort of works. So here we have a couple of regions I switch to this reading it on actually color being updated.

Luca Marconato: You can find more detail in the website. But since I'm short of time, I will jump directly to this slider.

Wouter-Michiel Adrien Maria Vierdag: Here now…

Luca Marconato: So we want to thank all the core developers collaborators is an external contributors that helped in designing and…

Wouter-Michiel Adrien Maria Vierdag: if I switch again to another class you also see the color in the viewers being updated and we can draw it then adds to layer those features,…

Luca Marconato: implementing this project and I thank you for your attention.

Wouter-Michiel Adrien Maria Vierdag: which is automatically if a user safe stats is parsed into an end data table and then starting the objects.

Luca Marconato:

Wouter-Michiel Adrien Maria Vierdag: Now to the transformations in aparri. So this initially is more from a user site perspective.

Luca Marconato: exactly

Wouter-Michiel Adrien Maria Vierdag: So in a party every layer is ultimately inheriting from the base layer and this base layer has different kinds of Transformations living as an attribute there and those different attributes are a fine rotates skills here and translate which a user can directly provide in order to change the coordinate mapping in the back end.

Wouter-Michiel Adrien Maria Vierdag: Just to go back here at the moment. We only support these linear Transformations that you see here internally.

Luca Marconato: Yep.

Wouter-Michiel Adrien Maria Vierdag: If you provide one of these it's automatically converted into a fine as So Internally and that's the point that I'm getting here to this is sort of what it looks like the base layer has this privates. Property that is basically cell transforms, which is a transform chain. We have here the different current space because we have multiple things that we actually interact with. So we also have the data coordinates itself to think of and in certain cases if you want to visualize multiple layers also those currents of visualizing side by side.

Wouter-Michiel Adrien Maria Vierdag: and most important for the user and what actually every attribute on layer that I just showed in text with is this fiscal 2 world current system, which is an extra transform applied in World coordinates that typically lines one layer with another And we also have the initial other layers that I mean are more important when you actually have metadata as well. So the physical coordinate system that may encode for example the acquisition parameters. So what kind of resolution you're actually working with and with what kind of units and that's more this part. We also can work with multi-scale data which tell to date is quite important.

02:20:00

Wouter-Michiel Adrien Maria Vierdag: With the multi-skilled data, we also basically keep on checking what part of the data is actually being shown directly in the viewer and at what zoom level which also then if it's represent this multiskilled data and not to have too much data being sent to the GPU and overall keeps the amount of memory being used at any given time very low.

Wouter-Michiel Adrien Maria Vierdag: What this ultimately looks like a set before the different attributes in a party all interact with this transform chain.

Luca Marconato: so I'm sorry.

Wouter-Michiel Adrien Maria Vierdag: And as you can see it all interacts with this data to physical the other parts are not directly available to the user and as you can see, we also don't support at the moment long linear transforms and I will talk a little bit more about that later. But this is the basic interactions at least on the price side. And I actually don't show slides at the moment of what does this look for the rendering part of nepari, but I can briefly mention it. So the rendering of nepari currently is being handled by this pie and fespi has this transforms module where ultimately you just work with shaders that provide coordinates mapping and it provides typically both the mapping and the inverse mapping at the same time, which you can then access them basically work with

Wouter-Michiel Adrien Maria Vierdag: And one example of what it looks like at least for an apprised patient data given an example with Excel and vision data. So here at the moment you see that we have data in the global current system that's not lined. And we also have it living in the line space and you see that the fine transform here is automatically updated. This is more like the visual example and ultimately what it looks like in the code, which I mean.

Wouter-Michiel Adrien Maria Vierdag: Look at also showed part of the API for but this more like an Empire spatial data directly. Is that We update basically current system widget, which basically gives you then which coordinate system is selected and we check with the Transformations for the particular ement. What kind of Transformations are available. We retrieved a particular transformation. Of the current system for that element and we just converted to a fine Matrix. So ultimately the price spatial data just into x with a fine attribute of the base layer.

Wouter-Michiel Adrien Maria Vierdag: And I mean with that there was the presentation and I would like to thank the whole Nepali team also the funnest challenge perk initiative and…

Matt McCormick: at least Some time advance and…

Wouter-Michiel Adrien Maria Vierdag: next to that also, of course the whole spatial data team.

Matt McCormick: time one question maybe to this idea.

Wouter-Michiel Adrien Maria Vierdag: Look at already has done the Shameless advertising…

Matt McCormick: We had of taking the spatial data model.

Wouter-Michiel Adrien Maria Vierdag: but not entirely we are also available both for an apari and…

Matt McCormick: Is there any interest or Be possible to use that more.

Wouter-Michiel Adrien Maria Vierdag: SC first on so lip in case there are any questions SC first also as a discourse and…

Matt McCormick: directly in the party spatial data or nepari

Wouter-Michiel Adrien Maria Vierdag: I didn't put the schedule here for special data,…

Matt McCormick: or no

Wouter-Michiel Adrien Maria Vierdag: but spatial data and the part. We also frequently old Community meetings. So, yeah, please feel free to get in touch if you want to contribute work together, and I've also done some parent coding sessions with nepari. And happy to work together with anyone that's interested. Are there any questions?

02:25:00

Matt McCormick: thank you so much. Yeah, that was very very nice. I'm John did you want? start

John Bogovic: Yeah, Thanks I'll share my screen in a second. just move stuff around.

Wouter-Michiel Adrien Maria Vierdag: And I mean to directly.

John Bogovic: Okay. Hopefully you see something. Yep looks good.

Wouter-Michiel Adrien Maria Vierdag: Use the module from a page today.

John Bogovic: Yeah looks good, I guess.

Wouter-Michiel Adrien Maria Vierdag: I mean at the moment the current API we are directly using in a very special data pretty much to have to transform.

John Bogovic: Great, so I'm gonna talk about big warp today. So I guess let me introduce myself in case people came in…

Wouter-Michiel Adrien Maria Vierdag: The only thing is that with the mapping for the rendering that's all happening on this by site.

John Bogovic: since the beginning but my name is John bogovic. I'm part of the saalfeld lab at HDMI engineilia.

Wouter-Michiel Adrien Maria Vierdag: So What is happening on the website is that we really create the Shader.

John Bogovic: Stephan saalfeld has joined the meaning and a minute ago, I think so.

John Bogovic: I don't know.

Wouter-Michiel Adrien Maria Vierdag: And that is the main thing…

John Bogovic: Do you want to introduce yourself Stephan? Okay, that's no.

Wouter-Michiel Adrien Maria Vierdag: where I don't think that this is the module that will directly provide and…

John Bogovic: So big work is a plug-in for Fiji,…

Wouter-Michiel Adrien Maria Vierdag: the mapping of that site for the rendering.

John Bogovic: which is a Java based tool for interactive deformable Landmark base 2D and 3D image registration before I get into that. Let me just show some of the background tools that it depends on and I have too many windows open. Let's see.

John Bogovic: So this is the Wiki page for big warp on the image Shay Wiki. It depends heavily on image lip 2 which is a Java based tool that's essentially a multi-dimensional image processing library and a lot of the lazy processing and just handling of data that big work does critically depends on image too the image viewer that big work uses called Big Data viewer Big Data, you also depends critically on image look too and both these projects biggit. If you were an image of two are things that our group contributes to and uses really a lot.

John Bogovic: So my intention for this is to make it at least 50% hopefully more live demo, but I'll just start by showing you this which is the big warp interface to windows open one is going to contain you're moving image or images and the other is going to contain your fixed image or images and the way you interact with it is by clicking Landmark correspondence pairs. There's going to be a table that shows you all the landmarks. ed you hit a hotkey and then that applies the transformation to the moving image and then you can update it kind of on the fly something. I won't emphasize a lot today, which is very important. Is that big warp and Big Data viewer from the start really we're designed for a huge image data. So one of the first projects that we usually work for was

John Bogovic: This huge electron microscopy image data set of the fly brain here that was collected at chenelia. So it's I mean approximately a hundred terabytes on disk and while we're not usually applying transformations to the hundred terabyte image data set. We need a viewer that can visualize those huge multi-terabyte image data sets because those are often the targets to our image registration. though, we can if needed apply transformations to these huge data sets using these methods on the Fly and ndff and omizar are now critical and what the format that we're now preferring for storing these large data sets. Let me hop into the live demo and I think let's move this way. So This will also come later.

John Bogovic: But here's two fly brains that I've opened. These are kind of small-ish images. These are going to be in the hundreds of megabytes. I think range as every sample them now you'll see if I've clicked a bunch of points here. I can also just make these points bigger if you can see them.

John Bogovic: So there's a bunch of them that I clicked and if I can just overlay the images and you'll see right now, they're not nicely aligned. But if I hit the hotkey it applies that the transformation and you can see they're nicely aligned so you can also on the Fly move these things. So, this is a thin plates blind transformation right now and I guess things to emphasize are that this is really lazily being applied this transformation. It's not rastering this entire image. It's only applying a transformation to the points in the image window that are being requested as a drag this around. So another thing you can kind of see is if I move this around actually, let me do it the other way.

02:30:00

John Bogovic: yeah, I mean, so here you're seeing sort of two different views into the same virtual array and they're just being updated line as I drag this point around and that's the basics. I think most people here are familiar with timid registration and I don't want to go into more Basics rather. I'd want to talk about How big warp interacts with the specification that we've been working on and I guess there' I want to show I think there's two main things that I want to show and they have to do with how images Transformations are exported and imported. I guess the first thing I'd like to show that there are a number of Transformations types that big warp can estimate for you. I'll switch to affine now, for example as you see you I move this around it's constrained to be an affine transformation.

John Bogovic: Exporting Transformations is relatively easy. this has a lot of options. But the main thing here is this sort of ngff format type. and the reason that's important is because It defines the format and the specification that big work will use to export transformation. I have this Zar container here. And so I'm just going to paste in that name and I'm gonna just say I want please export this current affine to this location and if I say, okay Here's that folder. You'll see that there's an affine there and if I look in there.

John Bogovic: You indeed see that there is an affine that will save there. This looks quite a bit like the specification that we're working on this we don't need. So this has been useful for us in terms of standardization and something that I also think is very important and that'll talk a lot about is these Imports this input and output tags and you'll see that they relate to the names of the images that I'm working on here. So it's gonna be probably small for you. But the names of the images are displayed at the top of these windows and indeed those correspond to the input and output coordinate spaces that this transform knows about I'm gonna export this transformation in one other way. which is now if I have this thin plates blind it's going to be of course the deformable transformation and

John Bogovic: By default if you use this ngff format for thinkplate spline. It's going to export it as a displacement field since that's the most interoperable format. We also have this tag for this TPS format, which we can see an example of later perhaps but first, let me show you what this might look like. I hope I still have this copied. So all into that same as our container. I'll now make a detailed. I'm Gonna Change. These parameters slightly just so that it gets exported a little bit more quickly. I'll hit OK and then hopefully in a second we'll look in that container again.

John Bogovic: Yep, there's the B field you see a bunch of information. the blocks of those are container. Here's the information's at least the shape of the array is what you would expect. This is going to contain the Vector components also important. This is a coordinate system that defines the grid that the displacement field was sampled on so there is an axis that indicates that This Dimension represents the displacement vectors and as well, we have scale and Transformations which Define the pixel to physical Court transformation of the displacement Fields grid so because I specified please use four Micron resolution. That's why we see for there. I heard a little notification.

John Bogovic: there's not a question is there?

John Bogovic: I guess I'll continue. There's one other thing. I wanted to point out which is the dislace a representation of this displacement field coordinate system was also saved to this location and here what this is saying is that there is a displacement field there is a transformation of type displacements and it's located at the path dfield inside of this container and it input coordinate system is this one and it's output current system is this something that one could do now is point big warp to this coordinate system.

02:35:00

John Bogovic: now it can apply so it will know please load the displacement Field located here and apply it to the image that I've pulled it to and all works. so I think that's an important way in which big warp interacts with these. specification I'm gonna move on something else but now is a good time for questions if there are any English

John Bogovic: Okay, there is a race hand. Using the dot Dimension separator for Zara is a bold move. These images are tiny. the default is now changed but yeah, it's sort of user specified but

John Bogovic: anything else

John Bogovic: Because if it's not and I use for example nearest neighbor interpolation on the displacement vectors and you use linear or cubic, then we will get different results. So to actually fully specify the transformation in a continuous coordinate system. You have to know how to interpolate it or how at least I interpolated it so that you can reproduce my results.

John Bogovic: That's the really good question.

John Bogovic: All right, cool. yeah.

John Bogovic: I'd say that it's I don't know. Yeah error, perhaps it's at least. willfully ignoring useful information in a way that makes your results not reproducible. I know. I'll let looter and Stephan decide who goes first.

John Bogovic: Who do I make you go?

John Bogovic: 100% you can apply a displacement field to an image of a different resolution easily. The default is usually the grid to the field of view and the resolution of the target image. That's a pretty typical choice for the Grid on which to sample the displacement field. Another common one is we estimate the field of view of the Warped moving image. And that's a useful thing if that's much smaller than your target image. For example, you might Have a ton of zeros in your displacement field if your displacement field really only needs to be defined over a smaller area.

Wouter-Michiel Adrien Maria Vierdag: I thanks.

John Bogovic: So those are probably the two most common, but also great question.

John Bogovic: your return Stef

John Bogovic: Cool, that's great. I'll continue on so let me see how yeah, okay. I definitely want to continue on because I want to show this so next I'm going to show a case study which really relies critically on the fact that Transformations know their input and output coordinate systems. So in this case study let me see we have four different images and I've color coded the magenta green cyan and yellow and it's not too important what they are, but the fact is that they are all of exactly the same sample. In magenta, we have an electron microscopy image and this is the smallest physical field of view, but it's the highest resolution. This will be collected I think. Eight nanometers possibly four nanometer resolution and I think this block is going to be tens of terabytes in size on disk.

02:40:00

John Bogovic: So then a B and C are all x-ray tomograms which are in the hundred nanometer. So 10th of a micron to maybe one or two Micron resolution and have a much larger physical field of view, but are smaller images in terms of number of pixels. We'd like to register all of them. And right in fact, we'd like that we'd like to use the electron microscopy as the target image space, but I'd argue. It's pretty much impossible to register this C image to the EM directly. So rather what we're going to do is we're going to create a chain of transformation where we register to the electron microscopy then C to B, and now if we wanted to register C to the EM, we just have to apply.

John Bogovic: No first T3 event 2 then And so this relates a lot to a lot of the conversations we've had earlier especially with mohammadi being able to compose Transformations is critical and for example, if T3 knows that it starts at c and ends at B. Then we know from examining the Transformations themselves how to apply them. So I'm going to show you a live demo of this part and actually now that I'm going to show you is exactly this data set. This is open organelle, which is Genelia cell not projects team data repository. So the simple that I'm going to show you is a fly heart. It is let me tell you how big it is.

John Bogovic: So I guess it says it here Yeah, so this is in the dimensions in nanometer. So it's going to be a hundred and sixty-three-ish thousand nanometers at 8 nanometer resolution. This is going to put it at around 15 to 20,000 pixels per Dimension and I think I measured that it's about 2.8 terabytes on disk after compression. so now here's On the right you're seeing the election microscopic image that I was pointing to you on the website. So again, this is the 2.8 terabyte image and on the left you're seeing one of the X-ray tomograms that zoomed in to be approximately the same resolution and I would challenge To tell me if this is correctly registered to this or not. and I'd argue that it's impossible. And let me zoom out and to show you what this image actually looks like when the zoomed out. This is a sample of a mouse heart.

John Bogovic: So we have more than just this image here. we have a couple of others. Let me show you one of them. So I think hopefully I need to make sure that this is

John Bogovic: There's one more specifically. I want to get to here we go. So here's another image. So this is an x-ray tomogram that was taken immediately before the election microscopy and much higher resolution than the other one and here you can really see the vasculature and the fine structure much better and I'd argue that registering this image on the left to the lecture microscopy is at least possible. So I did what I essentially described there here. I have five images open that they're not currently registered. I'll show you a couple more of them. Actually. There's the one.

John Bogovic: As I zoom out. I hope we'll see another one because another one. Hopefully visible in a second. Yeah, there's two more that these were Trimmed an image or not again with the xiaoman then there's that orange one that I showed. So I guess first point is that these are not currently registered. The next thing I'd like to point out is that I manually made this chain of registrations between these five images and I saved them somewhere and let me just show you what that looks like very quickly. Here it is. So this is a Json representation of a thin plate spline transformation. And again it knows What its input and output coordinate systems here's and here's an affine. It knows with its input and output coordinate systems here's another affine. Here's another affine.

John Bogovic: So what I've done is in big warp here. I've pointed it's gonna be impossible for you to see this but this path points essentially to that It parches that Jason file and here in this drop-down. It tells me all of the coordinate system that it knows about. And so if I select the electron microscopy image coordinate system here what I'm saying is hey big warp, please find a transformation for every open image. That transforms it into the space of the election microscopy image and if it can find it it will apply

it to all the open images. So I'm going to do that. And fortunately this image did not move and that's because the identity transformation is the one that would move it there. But if I turn on oops, I need to do a couple things. So first if I turn on fused mode now, you see I'm gonna just make this full screen.

02:45:00

John Bogovic: you see that magenta image and indeed it's well aligned. there's the other. So now if I turn on the green image, which is another trim. Oops.

John Bogovic: And the sign in which these are also well aligned. And there will align to each other and if I turn on the orange too, you'll see that that's a little aligned I'll turn off some of these other ones because these are five images so it gets pretty crowded. But yeah, and this is great because I don't have to do the tracking of What are the Transformations that I need to apply to any image? And in what order and in what direction and this is been? Great for me since previously I would have to do this either in my notes or in my memory and that's very unreliable. So this is been very valuable for Any questions about this?

John Bogovic: Okay, great.

Matt McCormick: is this using the latest version of the spec

John Bogovic: Yes, a I say yes. It hasn't changed in a while to the spec hasn't but yeah, I would say yes. The only thing that is not in the spec is this transformation type? So this is a bit Homebrew and this is something I had been intending to add for a long time and was hoping

John Bogovic: I think I'd like to get this version of the spec out and then add this type.

Matt McCormick: Does that correspond do you know to any of the IDK spline? for

John Bogovic: Yes, since I modeled my spine it's in place flying implementation after that then played it case then plate spline. So I expect it does yeah. and if it doesn't in any way I would like it to

Matt McCormick: okay, cool.

Matt McCormick: I think router and Davis had other questions.

John Bogovic: yeah, that's let's go.

John Bogovic: Davis why don't you go first?

John Bogovic: Yeah, probably show the transforms you can show and then I found something that I don't understand and I don't know what to do with this. And I don't know if I'd recommend a fallback or I mean, I think if a fallback is possible that makes sense, but I just don't understand if a client could know what a good fall back is if they just don't understand or don't Implement a thing. So that's going to be probably implementation dependent but Something like that. Yeah. Good question.

John Bogovic: Yeah true.

John Bogovic: I guess what are your turn?

John Bogovic: I guess the first thing to one needs to know is if one needs the inverse or the forward Trend displacement field, which is a challenge, but also we know that so this is the displacement field. We

need to apply. What I'd say is I'm gonna assume the polygon has some vertices. Then I'd say probably you're applying a transformation per vertex to apply the transformation to the whole polygon. So I'd say take your displacement field figure out what its physical coordinates are make sure the polygons vertices are in the same sort of physical coordinate system as the displacement Fields. Grid that probably gonna be the case. And then for every vertex.

02:50:00

John Bogovic: Interpolates the displacement field at that point to get a displacement Vector apply the displacement vectors the vertex and then you're going to transform her vertex point and then I think you're pretty much set. Is that clear or I mean I may also been missing part of the question.

Wouter-Michiel Adrien Maria Vierdag: Yes, thanks Finke a good idea. So regarding where to store transformation, so would you recommend to store a transformation between polygon and the image with them but it's listening to One take the same displacement field and put it to the polygon and if so included as a teaser need to shuffle it.

John Bogovic: Say the last part again.

Wouter-Michiel Adrien Maria Vierdag: Yeah, so if I take the email and I won't really drop payments because it's too big and unity. They want us to share the displacement field applied to the polygon to the collaborator because that's ultimately what he cares about. Can I take this and put it to a polygon as it is what we need to do something on it.

Wouter-Michiel Adrien Maria Vierdag: There is an interview the Rosa ethernet problem. We couldn't hear you the 10 seconds.

John Bogovic: Okay.

John Bogovic: I'm not sure. I think I'd have to think about that more. I mean, I think I'm just missing the important part of the question. I think that if the question I think what I would recommend is Apply the transform to the polygon assuming it's small enough and send the transformed polygon in its entirety to the collaborator, but I think I'm missing something. yeah, so Maybe we talk…

Wouter-Michiel Adrien Maria Vierdag: Okay.

John Bogovic: if someone else thinks they understand and can answer go for it. Otherwise, you can definitely talk online.

Stephan Saalfeld: Maybe I can give it a try. So the Transformations aren't bound to a particular image or…

John Bogovic: before

Stephan Saalfeld: anything There are independent. So you can store them in an arbitrary place in horrible in an hdfive container and you can reference them from an image, for example, so you tell a big part open this transformation you can use the same transformation to transform a polygon. So you wouldn't connect it to a particular element. and for one thing this is actually super important as John just told you he has this storage of several Transformations going from this space to that space and from that space does it this space and so on? And you can reuse these Transformations for many images by just referring them.

Stephan Saalfeld: With the URL that points to that particular attribute in a container tree. And that is particularly important.

Wouter-Michiel Adrien Maria Vierdag: okay.

Stephan Saalfeld: So we have other applications where for example you have very many tiled images. And you know that all these tiles were acquired by the same confocal microscope with the same lens Distortion that you have estimated using a dedicated protocol. And you don't want to store this lens Distortion transformation for every individual tile again, because that's stupid or you just want a reference to that particular lenses Source transformation. So in your chain of Transformations that are using and applying to every tile you can have a reference to a transformation that is stored elsewhere or in the same Tree by your LD references that particular attribute. Right and you can apply the sequence to all the tiles and you don't have to replicate the same thing. and then I wanted to answer to the question that the proposal that came from Matt's room. That's several people in Matt's room. Is that right?

Stephan Saalfeld: Anyways, cool.

Matt McCormick: yes, but it's

Stephan Saalfeld: So I think that's a very good idea to build a service that can convert as many Transformations as possible into deformation fields. And we in fact do have reference implementation for this locally in based on image of two. And what this implementation does it generates chunked deformation Fields so it can generate an arbitrary chunk. that is generated from an arbitrary chain of Transformations as a deformation field. And that chunk can then be cached and used to transform extremely large images. So you're basically don't have this running out of memory problem. Because you're generating relatively small and compact deformation fields at a time. And then you can catch them or store them or send them over the Internet or something.

02:55:00

Stephan Saalfeld: In big Corps / image lip to be just immediately use them and when we're running out of cash, we just reproduce them when it's necessary.

John Bogovic: Yeah, good question. We definitely have used multi-skills for deformation fields. the way in which we use them sort of varies per application. So the specification.

John Bogovic: I wouldn't say it directly represents a displacement field in a multi-resolution way one certainly could save the same deformation field at multiple scales.

John Bogovic: This is a great good idea. And maybe it should be more explicit about it. Or this case is where it's useful. Yeah.

John Bogovic: Yep, and over time I can show maybe one more thing possibly for just as a teaser in case people want to talk to me about it offline another time. So let me get rid of this as a teaser Let me find a slide. and then I'll be done. so

John Bogovic: so we have this idea for applying transformations to masks essentially. or let me put it this way. I need to go back a couple slides one. Second. Things are a little slow.

John Bogovic: no, okay. I don't know why I can't just So in this application, this isn't just a toy example. I have this boat.

Matt McCormick: Thank you, That was awesome. So now we have

Matt McCormick: I see Nick Nicholas here to talk about an sex and most registration over time. And so we had scheduled a half hour. We're a little bit behind but feel free to use the whole half hour. So after this, we have a break for lunch for the folks on the East Coast the US.

Nick Tustison: No, no, I mean to be honest Matt. So when you asked me to give the tutorial I was kind of going back and forth between 15 minutes or 30 minutes, but it seemed that the universe is spoken that says the only one here 15 minutes from me and I'm totally fine with that because as you know, I'm on vacation. I'm looking at a lake. so 15 minutes is great.

Nick Tustison: so I put a link in the income messages.

03:00:00

Nick Tustison: I just put some notes together us about this tutorial if people want to follow along. let's see. So now I want to Okay entire screen. Okay, let's see. Okay, perfect. so as Matt said I'm kind of one of the ants Developers.

Nick Tustison: but as I was putting this tutorial together, and trying to focus specifically on the subject matter, a two thoughts came to mind. in that a couple potential transforms that might be of interest that people might want to

Nick Tustison: hold on. I'm getting a message. Okay, and the people might want to consider are in ants we use What's called the symmetric normalization Which is an invertible transform, but it also creates a virtual mid space. And having the transform to that mid space might be useful. and also didn't see mention of velocity field transforms. and that might be a consideration. I'll talk more about that in a little bit. But just so people don't think that I'm here to sort of plug for my own stuff. I want to emphasize that I'm ants is built on itk. and a lot of the stuff

Nick Tustison: Basically, all of the stuff that I'm going to talk about all of the hard work is being done by ITT filters. And so if anybody wanted to do a sort of a simple idea Port one could easily do that with what I'm gonna talk about today. in order to demonstrate the application of what I'm going to show I recently wrote paper with several colleagues dealing with mapping the mouse frame. and it's particularly interesting use case. with regards to spatial transforms

Nick Tustison: in that people at Penn State developed. no, I'm going to discuss that later. but involves the a velocity field transform Okay, so people are aware. all of the

Nick Tustison: in addition to all of the code that we wrote. there's

Nick Tustison: So here's the ecosystem tutorial. and basically it covers the ants X code base, which has a command line interface and then in addition to the code base.

Nick Tustison: We also have an r and a python interface. And let's see. So there's all of this functionality that we have and regarding the Last talk we also do a lot of landmark registration. And so we have this particular function which takes a set of numpy or our arrays, comprising landmarks and you can do and

we have several different transformation types. Yeah, and they're listed here rigid similarity affine a Vine transform type A thinflates Fine an exponential different symmetric normalization and a velocity Very transform type.

Nick Tustison: And we give examples you can run them. And all these are self-contained. So in examples that require images, we provide the images sort and interface to actually download images which are stored on fake share in this case. These are just simple points.

03:05:00

Nick Tustison: Let's see. Then. Here's a more complicated example where we're composing transforms a rigid plus a symmetric normalization transform. And this is demonstrated here.

Nick Tustison: And then let's see here we have. and then in addition we have sort of velocity field More than two Landmark points. So in case one has a temporal series of snapshots that they want to sort of Link together with a single transform. That's available and of wait, All of it uses all so these filters so the velocity field is stored as an itk image. And one can then and then what integrates that velocity field or that four-dimensional image?

Nick Tustison: To create a displacement field between any two time points within that velocity field. Okay, and so, let's see.

Nick Tustison: so one of the things that's nice about these transform types it is that they can be weighted. So for example, if one wants to wait boundary points more than other points one can specify a greater weight for that and we have examples showing that so let me actually go to the application. wait, no. Sorry.

Nick Tustison: Let me and talk a little bit about our deep learning framework. so in addition to our and python interface or ants we have deep learning libraries. kind of analogous to manai that are extensions of there are and python based extensions on the underlying deep learning code is A Keras and it's not for any preference for Keras over pytorch. simply due to the fact that at the time Keras was more developed in than torch was? Okay.

Nick Tustison: So for our deep learning let me go back here. so, our Ants xnet is shorthand for ants Arnett and ants pinet. And we have a bunch of different applications. So we have brain applications. We have learning applications because I do a lot of loan work. We have Mouse applications. It's part of the recent development with this paper. I was just talking about General applications and then other stuff such as data augmentation, which

Nick Tustison: we think it's pretty neat because we don't see it in other packages such as histogram intensity warping simulating bias field kind of an inverse of sort of the well known and for bias correction, and then since we do and since we built a lot of the transformation work with itkv4, development. we also Incorporated that into our random

Nick Tustison: So here's the paper. it's currently on bio archive. And basically involved our work with Penn State. and Us in particular cute Let's see.

03:10:00

Nick Tustison: okay, so this is so put together an entire GitHub repo. So that everything that's contained in the paper is entirely reproducible. So there are two applications. We do sort of a brain extraction and a brain presentation using only the Allen brain Atlas and aggressive data augmentation as I showed to be able to train this network. And then we showed it on an completely independent data set how well it works.

Nick Tustison: and that's all part of the ants pineapple and then what we did is we took these labels and we use them to create a velocity field transform model. from What's called the dev CCF? but a mouse brain Atlas and this involves what it was is the folks at Penn State. taking seven developmental Mouse brain stages both embryonic and post birth and a brain stages and the had I think around either 10 10 to 20 subjects. for each developmental stage

Nick Tustison: acquired multimodal images for each subject for each brain stage Okay, and they used ants tools to create multimodal atlases or each brain stage and then what they needed to do was they wanted to link them all together in a single transformation model. so that's what I'm showing here. These are the a images? of each of the seven brain stages and we used these tools that are available in ants pry and ants are

Nick Tustison: You created A lot to creative a velocity flowmought transformation model. It's basically an itk of 4D itk image with Vector components at each pixel. Okay, and so all of the code. Is here this is a preliminary example where I do sort of a simple polygon example where I'm working a square to a circle then back to a rectangle or something like that and I link them all with a single threat velocity field transformation model and show how you can warp to eat any space within its time domain. and that's just a preliminary example that runs really quick just to make sure that you are. Running the code correctly. And then all of the data.

Nick Tustison: You reproduce the velocity of transformation model for the dev CCF. Atlas is here in the repository. There's a link to this shows the data and this basically breaks down how the steps necessary to create this velocity transformation model and it's already in Python. I could have written an R. But python was just easier and so here using regularly registering all the images to want to one of the last as sort of the domain image. And then here what we do is we perform a paralyzed registration.

Nick Tustison: Of all of the images to be able to get the correspondence and then those constitute our landmarks that we then propagate to all of the atlas spaces and then we use the landmark based velocity flow. Transformation model building do you actually build a model? I only actually show how sort of convergence happens. here's some of the examples. So here's where we then take the single velocity field transformation model and we work every Atlas every time point to every other time point. And then another possible utility. is

03:15:00

Nick Tustison: Three. it's here in the paper. last thing I'll show finish up. There's another example is suppose you wanted to create pseudo templates, right in between developmental stages. So for example P4 was one of the existing developmental stage and p14 was another existing stage. And so using the velocity flow transformation model we can Warp. each

Nick Tustison: Can't each of these existing developmental stage images do the space of and this is an arbitrary choice of P 10.3 and use those two images and to create a pseudo template from those two images and here I show right in the case of T20. And again, this is all sort of publicly available. All of the

data is there and anybody can reproduce these results? But just kind of shows perhaps sort of we might want to consider a couple of these other transformation models in sort of this new developments Within itk. Okay, it looks like I'm done. Thank you.

Nick Tustison: Anybody have any questions, sorry.

Nick Tustison: But for what?

Nick Tustison: Right, so I don't know what the interface between simple itk and money. I haven't looked into it. But the way we kind of Envision the ansex ecosystem. Is that ants are and as our net are tightly integrated, right? So as our net contains all the Deep learning stuff, but as R has all of the transformation, doing stuff like transforming images or stuff like that so we don't.

Nick Tustison: No. No, I mean and a lot of people have asked us about that, about hey, why isn't there deep learning in ants and we talk about this all the time. our sort of thoughts on that is that registration all the stuff that's available in ants and ants registration and all the work that would be required to take that over the benefit kind of out ways to cost at least for us at this point.

John Bogovic: Yeah, I guess I had one…

Nick Tustison: If that makes sense. Okay.

John Bogovic: if there was another one when you're making these pseudo templates, for example, do you have just a feeling for how finally you have to sample the development in order to make I'd expect for example that you to get a good pseudo template. You should sample in time where development is happening fastest or something like that.

03:20:00

Nick Tustison: Yeah, so I'll be honest one of my so I developed this velocity flow model, for landmarks. On prior to the dev CCF work. It just happened to be that they came along and I had this already built not for any in particular just because I thought of being interesting idea, So when I showed them we could sort of Link all this together. They were like, wow, this is great and then they started playing around with it. and they specifically said What we're seeing is the pseudo template the prediction of specific time point.

Nick Tustison: It doesn't Accord with what we see in the hippocampus is what they were looking at specifically so we can use this to kind of look and see where we need a sample. so do I have sort of a theoretical idea of

Matt McCormick: So there's no registration in the ants X in the Charis Okay.

Nick Tustison: how finally these need to be sampled? No, I mean I was

Nick Tustison: It's just starting to get played with us. and one of the other interesting things though. Also, I'll mention about sampling is it the dev CCF is nonlinearly sampled in time? Right, so I kind of had to do some stuff where I am. You didn't kind of a long transform of the time points which made it a little bit more evenly space but it still wasn't exactly evenly spaced…

John Bogovic: Yeah, that's correct, and it would be great to have.

Nick Tustison: but this particular framework handles that non-even spacing.

Matt McCormick: Yes. Thank you. Yeah. Sure.

Nick Tustison: Thanks.

Matt McCormick: Any other questions?

Nick Tustison: yeah, and

Nick Tustison: What no I was gonna say is it? I would imagine. Yeah, and I don't know the complexity as I'm sure there are right. I just thought people should be aware. and one of the other things is it even though sin symmetric normalization is available in ants and available in itk. We don't.

Nick Tustison: Because it's so easy to work with displacement Fields. it's just a subset of displacement field transform. That's the way we designed it. But there's several interesting use cases where what you want to do is take advantage of this middle sort of space that sin uses and you want to be able to have access to that transform to that middle space in between two images. And this is something that our ants are an ants pie interface permit that itk does not because it wasn't easily sort of integrated into that design at the time.

Nick Tustison: It might be so ultimately before it sort of reduces everything to the forward and the backward displacement fields. Often it's optimizing for displacement Fields. So the forward and backward from the fixed to the middle and then a forward and backward from the moving to the middle. And so it's only four displacement Fields. They're being optimized and then right at the end. Before it kicks back to different finishes it sort of just composes those transforms.

Matt McCormick: In the current. transforms proposal.

Matt McCormick: There's not a velocity transform. Is that right? John

Nick Tustison: And…

Nick Tustison: then returns that to the user. And so yeah, there's some way to sort of expose that or make that available. Yeah, that'd be great.

Matt McCormick: And it shouldn't be too difficult right to add that.

Nick Tustison: Yeah.

Matt McCormick: It's hard to chug.

03:25:00

Nick Tustison: Yeah, that'd be great.

Matt McCormick: Do you think it?

Matt McCormick: Could we effectively implement? The sin as a series of three chain displacement fields and store it in an omnis are that way? Maybe

Matt McCormick: Thank you so much, Thanks for taking the time on your vacation. Really appreciate it. so maybe we could do saving for displacement Fields is effectively what we need and then maybe adding us specific type to that and we'll have sin support effectively send supportive.

Matt McCormick:

Matt McCormick: very cool.

Matt McCormick: Outstanding at great discussion. So let's take a little break. And wherever you are in the world. We're gonna have some lunch here in the east coast and we'll continue back at the end of the hour. Wherever this is for you. And we'll have a few more tutorials Hastings talking about his work on Transformations andras. He's back gives some insights into how Transformations are using a vtk. and then we can maybe continue on with the discussion of the discussions we had earlier.

Matt McCormick: I'll leave the meat open and you're welcome to use it and discuss and the chat too, but I will see everyone back in about 50 minutes.

04:25:00

Matt McCormick: So Hastings is going to get the next tutorial.

Matt McCormick: Just making sure join the right meeting. So it's the same one way around.

Matt McCormick: What's that should be? Very extreme it is. not

Matt McCormick: nothing yet. Okay, that is.

Matt McCormick: Absolutely not.

Matt McCormick: I'm ready over all I'm going to be opinionated today and present why your registration Library. should output an itk transform And the good news is that I might be. preaching to acquire here and that you guys have made it looks like already the beautiful decision.

Matt McCormick: To not do this where you write an algorithm that works on a specific resolution. You can register it beautifully.

Matt McCormick: when you are given new images to register you resample them to the resolution where Align, the resampled images and declare victory.

Matt McCormick: So we can look at this specific case is pretty common to be honest. Where You see for example. not isomeric inputs without it was spacings that aren't to power of two where you're neural network throws up unless it's a power of two. It's pretty common. It was definitely the beginning of a PhD. I think a lot of some of the early Moon I work where We can say the images are registered and the downstream MD PhD looking at their tumors says I have no concept of what to do with the stencil. You've emailed me. so there's two sort of. big problems here Which is that? The downstream code the tumor resizers of the world.

04:30:00

Matt McCormick: already are sort of having to deal with spacing image orientation and major Origins So they already have opinions about how to solve this. And in addition in the Deep learning world, we have

sort of internalized this principle from a unit that we can leave it as their problem. Because you can sort of fundamentally resize your image. Segment it with a unit and then send it back to the other guy in a way. That's really intuitive. they can you just undo the resize and that really fundamentally doesn't work for a displacement field or an affine Matrix or whatever. and so it's created a little bit of a gap between the worlds of trans Moore than brain Moore and then the clinic where everyone's still just uses ants

Matt McCormick: so the solution comes down to two priorities number one. Just don't output your transform as an annotated met unannotated map. So this is I don't know if you guys have heard of learn to Reg. And the sort of machine learning registration world, but they're the center of this problem where they've made registration really easy to get into by defining the problem where the correct output is an annotated tensor of pixels of displacement vectors measured in pixels. and the result is that it's gone not seen these brilliant methods coming out of it that get very high performance. Haven't seen widespread adoption.

Matt McCormick: For people who can't quite measure in pixels and then the second sort of lower priority once we've decided that we're going to store the metadata needed for a transform to be usable. with various image resolution and spacings Is what if we could coordinate? And that's a beautiful dream. but the sort of shelling point that is available suddenly. Is this itk transform? so good news is we have IDK image and all of the file formats that it supports and it sounds like talking to Matt ngff has joined this universe of

Matt McCormick: Basically, we have imminent orientation spacing An Origin. along with our image

Matt McCormick: that means fundamentally an image becomes a function from 3D space to intensity. Because once you have all of that. and so this while there is this widespread like interoperability between mermal dicom ngff.

Matt McCormick: Mhd we don't have interoperability between our transforms yet. And so a lot of transforms are at level zero of interoperability where they don't have metadata and so they never could interoperate. But now we're finally getting into this beautiful world world where We have at least two. fully fledged properly metadata transform formats the ngff and itk transform that one could conceivably write a library to translate one to the other. Which is a huge step forward and that's going to give these composition warping images transforming points.

Matt McCormick: So The Proposal that I came into this presentation with was that we just all agree if we can read an itk images and write an itk transform. Then we can at least talk to each other. Even if it's slow even if it's on disk. then it sounds like in GFF is at the point where we could coordinate on that instead or even Make them interoperate. how do I call Someone with a hand raised? It just said that that's Davis right Davis.

Davis Bennett: So is there an itk transform like Json serialization I could look at.

Matt McCormick: I have good news and bad news. So the reason I sort of settled on itk transform. Was not It was widely used. It's a collection of formats and There's one option is hdf5. Which is vaguely saying and the other is just sort of plain texts separated by new lines in a specific way. So that's sort of

04:35:00

Matt McCormick: It's sort of tragic thing. we're in the early days of the browser when it's implementation. is standard but the implementation was not written to a standard, but Matt sounds like he has the right answer here. So what I do is you can write out into this htfi format, which is obviously similar to tizarre. And there's a vs code extension for looking at visually hcfi files. And that's helpful. And then also we can just play with it the python API representation.

Matt McCormick: but that question was harsh enough It says that we're on the same page here

Matt McCormick: I think that you've got your finger on the pulse there.

Matt McCormick: Yeah, so the approach that I took was to just import the itk library.

Matt McCormick: We can also look at that. The Atticus software guide has a description of all the print of how Everything is parametized on what they mean? Yeah. so

Matt McCormick: writing to a transform is

Matt McCormick: not bad and the itk software guide Does a really good job of explaining how you would do that on your own if you didn't want to import the library? But the issue is reading a transform. Because there' as we heard from the ants talk these blind transforms velocity field transform symmetrically normalized transforms. It's a real zoo that's built up. And so when reading is a really strong temptation to just load the library. And then once you have this C++ object in memory, that is the transform you can ask it to transform a point. Or to warp an image and it will do that in the performant way. and you don't have to know that the end that the underlying transform is a composition of Beast blinds and such

Matt McCormick: But it does then tie you to this larger external Library.

Matt McCormick: Yeah. I think it's got good Java bindings. but nonetheless that's the phrase good Java bindings probably strikes fear into your heart and probably should strike fear into your heart. I'll put in Missoula the link to the documentation in the describes. what the data model is

Matt McCormick: Good, but yeah, so the nice things about. this data model is we've got support for this proper zoo of transform types And it's all. extremely heavily validated and tested So that you are going to get composition implemented. Both you the library producing a transform in the library reading the transform. Are reliably going to agree on the order which a collection of Transformers was composed. They're going to agree on interpolation. inversion and you can be confident. There's not going to be a step in the middle that's dependent on image resolution in a way. You didn't know about ahead of time and breaks your code when you see a new resolution. and it is nicely serializable through the format.

Matt McCormick: but as you said You don't necessarily want just the library that serializes it you probably. Would prefer to have a nice Speck. and so the sort of

04:40:00

Matt McCormick: My argument for why? You would want to do this someone from the perspective of developing a library. We can sort of follow the story of this icon registration package. so we've got

Matt McCormick: this Library pip install icon registration, but then admittedly does bring in dependency on hundreds of megabytes of itk just for transform serialization, but on the bright side suddenly it. gives you integration with

Matt McCormick: the I think ecosystem. So if you want to perform brain MRI registration or near my knee MRI registration then You can. import this code that allows you to use these trained weights written in sort of idiomatic torch and support the sort of scary. world of Origins and orientations and stuff that I suspect are scarier to the neural network people like me then to the OMG zar and mass brain people who are used to dealing with this stuff.

Matt McCormick: And so what's great, is that on once you're using this? sort of Black Box transform implementation. The user code is at least saying so when we're reading in our images these with itk and read is going to loading all of our metadata into memory and it's not going to forget to handle Origins for example, and

Matt McCormick: we have in our library of Icon registration. We've got this registered pair thing that then correctly Builds an itk transform given a neural network and a pair of images to import input into it and allows resampling using that transform in a coordinate respecting way. And there's the big upside here. Isn't that this code? Is that much easier to write than just doing it yourself with the Torch dot resample image that it's the mistakes that you can make here will cause your code not to run it In a way that you can fix instead of having mystery. orientation bugs that don't show up till much later and then when you're in that world of user code using a registration and algorithm or transform you've gotten from someone else. The key really is to resist.

Matt McCormick: Resist the urge to dig in and grab at the underlying data and talk to the transform through the interface. So if you do need to go from a voxel coord To a point in the image instead of talking to the orientation and Direction Makes Us. The image has methods for getting into continuous coordinates and to getting back and then when you need to move the point, just call Direct transform Point directly instead of looking it up and it's fast. It's+ coded. I know it's not as exciting as handling it correctly, but at least you're not trying to integrate someone else's velocity field in the same way they did and trying to get the same answer. And then there is complexity. And that's gets into the library code where?

Matt McCormick: Here because a neural network isn't going to know image orientation unless you're doing something horrible passing it in. It's like a extra feature it's so when you have to write code once to create for example this wrapper function But at least you have responsibility for that as the library author and your Downstream tumor resizers. can just import this and not have to think about orientation and spacing.

Matt McCormick: It does get hairy though. you got lists of reversed ranges of dimensions and you have to Make sure that you're using all these itk cup functions that are not capitalized like python functions should be. So I understand the concern. So these two functions are sort of a lighter leaner. option to have high torch backed Transformations that Ben opposed to manai or these coordinate spatial Transformations, so right now

04:45:00

Matt McCormick: GPU

Matt McCormick: what this code is is sort of Is the interface between our library icon registration? And the itk library handling transform metadata. So it's all fundamentally part of serialization. Yeah, so it's taking transform that in the torch world might be represented as for example an affine transform all the way to a velocity field or even like a multi-layer perceptron Nerf type nonsense. And we're just baking it

down into a displacement with correct metadata But then if you do this then subsequent image warps will happen on the CPU. using

Matt McCormick: IDK functions for warping images

Matt McCormick: one of the big lessons we learned in writing that we should just keep the transform.

Matt McCormick: That the getting out in and out of neural network space transforms which are fundamentally at fine. Really should just be stuck as compositions. with the neural network transform output, which is a velocity field or deformation or An MLP or whatever, so that's one of a lot of the early attempts to apply neural networks. to registration real world tasks instead of keeping the transformed store to the composition. They tried to modify the deformation field using the metadata appropriately scaling and shifting it. and swapping the channels around in a way that depended on input image spacing and this slide is basically just a warning sign saying having gone there. I wouldn't recommend it again.

Matt McCormick: and it's better to just have a stack of transforms that you can nicely serialize and count on your Downstream consumers to load In a way that's in the same order that you put it in.

Matt McCormick: but basically the fundamental concept is that You want to by using this itk transform you add a lot of Gunk into your registration Library? and then With that pay and you buy improved reliability on Downstream tasks when someone who's never seen your code before is trying to run it on a pair of mouse brains.

Matt McCormick: serialization and DC realization is easy with modern itk that's by far the most Pleasant part of all of this is just throw into Pull it out of the file name.

Matt McCormick: so now dropping my opinionated nature of this is absolutely the right idea. There's pros and cons here. So the process we talked about are. Pretty strong and that you have apis you can trust you've got a standard. That's 15 years old. and so the whole ecosystem that already speaks this language in the slicer and itk side. And the medical people will be unhappy to see itk but we'll also know what to do.

Matt McCormick: One of the big advantages of this ecosystem. Is that If you serialize your transform as an itk transform and loaded into one of these viewers. Then and it looks right that lines up the images you can trust that you've serialized it correctly and that if you load it into another viewer, you will see your images worked at the same way. And sort of dropping that anxiety. That once you visually verified it and one of these tools that you've probably implemented this correctly is the pro of this approach, but the cons are severe.

Matt McCormick: So the python apis are clunky and capitalized and ways that make us uncomfortable. If you need these hundreds of megabytes of ick code to load it reliably. So it's really?

Matt McCormick: This PowerPoint was obviously made before I saw the earlier PowerPoints today and what I suddenly have is Hope. That this omes are project. We really can.

Matt McCormick: Get enough metadata, And not the implementation is working on it. that we can drop some of these cons by moving to the dot omes are and the nice thing is if we can put omes are as an extension that itk transform read and itk transform right can handle then. All of this code that I've showed already will continue to work with it.

04:50:00

Matt McCormick: although reading the spec for the transform would still be a great way to write on these are transforms from a library. It would also be really wonderful to be able to write them. by creating an itk transform and converting it or creating an itk transform object in memory and then serializing it to omizar. In then you get all the pros. Because you could see right you could choose what to serialize it to whether you need it to load into slicer. I mean eventually if you get it into slices and slicer could read it on these are so that's a beautiful dream.

Matt McCormick: And It wouldn't help with the second con though. The other big downside of the itk transform is that you cannot differentiate through it. it's fundamentally not Torchy. And so you're always going to need to have sort of two languages when you're speaking this. that's the case where Moon I has if Moon I can get a really strong transform standard. That has the correct metadata. It can absolutely eat its lunch here because of the performance boost and differentiability. I hope for that ending, but we'll see. Yeah, so in memory. Fundamentally a neural network that deformations can't work in millimeters. because it can't see the spacing. So if you're trying to register to adult brains.

Matt McCormick: Then if it was trying to Output millimeters, it would need to Output larger vectors. Than if it was trying to register to Baby brains that had been picked with spacing to look at the same size as adult brains. And so that's why the sort of translation that I talked about earlier is necessary and the gunky library code. So if you are. a grad student writing neural network papers to do image registration With neural networks, and you want this power of being able to ship transforms to Downstream papers? that they know how to process all sizes of segmentations with Icon registration. Skip installable And we've proven by fire that this works at this point where we've got this zoo of papers with a wide variety of internal.

Matt McCormick: approaches for representing transforms as big stacks of deformation fields or the stacks of affine transforms or in one case just a transform it directly outputs coordinates. slot into this same interface and can be compared against each other using and compared to Nifty and ants and all of the traditional registration friends So that's a zoo. You could join and suddenly becomes really easy to slap your registration method into and if anyone else is lung measurement or cancer measurement paper. And it doesn't involve extra code it's like five papers all share the same serialization and DC realization code. And it's draws Downstream users.

Matt McCormick: Which is I know that as a grad student, that's all I want to citations. But if you are most recent paper unigrad icon, that's much bigger. Let me close these really notes and zoom in.

Matt McCormick: so if you've got some images with metadata and you want to register then A pip install grad icon. Will grab you all code all of our weights. Beyonce I've torch working so install NVIDIA drivers yourself, but if that's hard which it is. It's not something we have much. Hope of getting away from but then the registration itself is just a single command line call. And works even though these images which are the standard sort of. Slicer test registration brain images aren't the Don't have quite the same spacing. You can. From the command line register them and produce a correct metadata transform.

Matt McCormick: So here's our before and before registration. And here's after.

Matt McCormick: and this sort of sales pitch that if you write your registration neural network. in a library with Siri let's metadata aware serialization of transforms you that we're seeing already people just grabbing it because it's fast to include into their papers. So this paper came out three months after. The

unigrad icon paper and it's already Incorporated it. And there are Varian cancer study. because they were already doing Nifty because Nifty is a beautiful registration approach and then they did. lock the morph because it's

04:55:00

Matt McCormick: it's the most famous registration approach for neural networks. So everyone's going to do the elbow grease to get that working and integrated into their resolution. but hey, they pulled in our paper too and got Some very strong numbers. So the dice of 0.82 for Nifty point at one for voxel Moore 0.93 for our approach and I think if we made it any harder to integrate then pip install and then one command line call to register. They would have probably gone with something that they've heard of before it says taking a chance on our approach.

Matt McCormick: if you support proper metadata, it adds to that. Appeal for Downstream papers to use your approach. so itk transforms are 3D slicer itk obviously ants there are Library icon registration and unigrad icon. I think I see I Tao elastic. I don't know if it connects to the serialization yet. So that one's a little spooky but Yeah, I mean. That being said again from what I've seen today of the level of maturity of the omes are transform format.

Matt McCormick: I think it might really be ready for prime time at this point my next step for the hackathon and make sure that IDK that transform right and write to this new format that is Json readable and won't be dragging 500 megabytes into your Java project. And still has these benefits. Handling spacing in a way that is correct and doesn't rely on a grad student typing. transform field times equals the image spacing one divided by image spacing too. Nice. Thank you.

Matt McCormick: questions and comments

Matt McCormick: laws, it's also good.

Matt McCormick: So I conclusion. by adding omizarre transform support to itk we'll be able to bring all those registration methods to all these other tools in the ecosystem. Yeah, that should create a sort of marriage of the two things the same way that by adding icon to itk the mermal and dicom because systems are married.

Matt McCormick: Slicer big warp. Yeah Dear special data, I think.

Matt McCormick: John bogovic

Matt McCormick: Yeah, are there any other questions or should we move on to that here questions or comment?

Davis Bennett: your muted jump

Davis Bennett: I don't hear you.

Matt McCormick: I think the end got or John. Did you have something?

Davis Bennett: John your audio is not coming through but it doesn't say you're muted. That's weird.

Davis Bennett: okay, John says I can go first so from a spec perspective And i t k has a lot of transforms and…

Matt McCormick: Okay.

Davis Bennett: I'm sure they're all really good. But the spec is targeting a lot of implementations and I think it would be good to find a way where if itk comes up with a Json serialization of all of these transforms. They could express that in omizarre without. Every other omes are compatible library and client needing to understand all of those transforms. So in metadata terms, that means that maybe the spatial transformation should be an extension point. That people can kind of hack around with and specialize the way they need. I don't think that's in John's PR, but I wonder if that is something to think about. because it's

05:00:00

Davis Bennett: it's entirely possible that everything that John has written is sufficient to express all of these transforms. That's the best case but if there is someone who feels like they do have a spatial transformation and it isn't. In the transform spec they're only recourse would be to submit another transform spec or just do it themselves and have technically invalid metadata. So I just wonder if we should be thinking about Poking a hole in the speck and saying this is an extension Point. Here's a bunch of stuff that we've already defined. And if you want to Define your own thing go for it. There's no guarantee to other clients will understand it, but you're free to express yourself.

Matt McCormick: so the place that I think we should start is have k write out standard omes are transforms. by keeping it as a composition as long as you're keeping at us Deformation as just affines and displacement Fields because IDK all of its diversity. Can be flattened so you don't want to flatten the affines? But I think bee splines and philosophy fields and all. Could certainly be converted to displacement field before passing before writing to the omes are format.

Matt McCormick: and then

Matt McCormick: yes, which sucks it's the wrong thing to do.

Andras Lasso: And I think please find transform is maybe like a hundred bytes and if you write it to displacement here at the 300 megabytes, so it's not really an option like to say okay homies are for that.

Matt McCormick: right

Matt McCormick: Okay, so we need.

Andras Lasso: You need to convert it to Great transform. But they're not that many IDK transform times. I think one thing why there are so many classes is that there are so many parameterizations which are useful for doing the registration. But once you are done and you just want to store it there is you store and find me tricks, but there are probably 15 different IDK classes that the engines art isn't a fine Matrix. So maybe you don't need a different storage format for each parameterization that IDK provides.

Matt McCormick: Yeah, so upon consideration and of these terabyte size displacement fields. probably Thin plane split thins plate spline displacement field and affine are sufficient to start with. right and…

Andras Lasso: And maybe and…

Matt McCormick: Beast blind

Andras Lasso: maybe be surprised so that that helps reduce so that would cover it. I think that at least for slicer and order a clinical Imaging application these classes like a fine B spline. It's in place plan and displacement field discovers.

Matt McCormick: Yeah is on big warp thin plate Splinter B spline?

Andras Lasso: the use cases

Andras Lasso: So both are used right for base Prime. We need structure points start to agreed while same place Prime can be randomly this through with it.

Matt McCormick: right cool That is correct and…

Andras Lasso: So both would be quite important.

Matt McCormick: more work, which is a common combination. But thank you.

Matt McCormick: John just

Andras Lasso: Okay, so I will be very quick and…

Matt McCormick: Yeah, Go ahead andras. Yeah.

Andras Lasso: What are the time constraints and I try to be maybe

Matt McCormick: You have this much time is as you.

Andras Lasso: And let's see.

05:05:00

Andras Lasso: Sure my screen.

Andras Lasso: Okay, can you see my screen?

Matt McCormick: Yes. Great.

Andras Lasso: Okay, so just the area that we work on with slicer and vtk. And IDK based applications is a mostly critical images ctmri ultrasound the X-ray and we always work in 3D space. So even if we have sliced images or art lesson images, they always live in a 3D space and often it comes with the time and color usually we don't have much other dimensions or any others and you also work with meshes, the segmented structures can be meshies or various stores and devices that are inserted into the body. And then the applications are surgical planning image Fusion your time surgical navigation. So anything that is not really real time, it's kind of like a limitation. So we usually need to have updates at least a few few have these per second or something I said these are going to interact.

Andras Lasso: It use these tools during an interaction during a surgery like being able to update the patient Anatomy defor in response to some sense or data or Imaging data.

Andras Lasso: and the transform time this is just that with this we discussed so in slicer we use vtk transform because it integrates nicely into the visualization pipeline, but I think it supports the same transforms and actual reviews the itk file I/O classes to read write or this transforms and if somebody wants to run apply these transforms and then either right here we can be used for that and we use the linear Grid or displacement field these brine same place Pine transport any addition to that we take us in recurs very ecology or whatever transform times but those are not use already this for the ones that we use

Andras Lasso: if you have a transform in I think we format slicer can read it and one application that we don't see often or one feature that you can transform it and there you can easily concatenate them split them invert them very easily using a user interface while I just have one more slide and then I show the demo so Yeah, I mentioned a few times that this vtk transformation pipeline has a couple of useful features that already working well tested and prove to be very useful in practice. So

Andras Lasso: that is our transformation types. They can be applied to both images and also geometries for any matches and also Anastasia reads structural grids surface and volumetric Ministry is point sets or that's already the formats that and it supports and in order to allow the same Transformer to be applied to an image or a me You need to have quick automatic inversion. because you need to apply modern transform to Transformer match why you need to resampling transfer to transform an image?

Andras Lasso: and this is automatically provided by vtk for non-linear Transformers warping transfers and BTK computes the inverter automatically by an iterative method.

05:10:00

Andras Lasso: and it uses the derivatives So the dvdk transform classes or always provide and the derivatives of the transform and I'm not sure if this is four years order requirements of being differentiable. And so if it does the same as the one I transformation want to do, but they may be the same.

Andras Lasso: And for immersion, we also need smooth sex operation because if we want to inverse them invertometries and at the boundary, we cannot just have the transformation go from some five millimeter Transit translation to zero because then you cannot compute the inverse vendors that this continuity so it also has a smooth extra population. So if you specify transform than it and reduces to zero as you get further from the domain of the transform, but it doesn't have an abroad Edge because otherwise you wouldn't be able to invert it. And basically Transformers also composable cyber show that you can concatenate any number of the transforms and you can also Transformer led to a concatenation.

Andras Lasso: Yeah, medical supervised laser resampling so you can put together its composed composite transform and you just I want to extract this specific plane from this image very where I applied five different transforms composite onto each other. So that is really fast. It doesn't need to apply the transform to a full volume because that would be extremely expensive but you can just be very at the specific point in space or playing smaller region and they told in the race with the VT pipelines, if you're visualization Pipeline and then you just put together a prime plan of input image. You can have private processing if there's any connected to that VK transformed and you apply some further processing creators and whenever you modify any data any transform any parameter anywhere the output is automatically updated and only that

Andras Lasso: 3 executed from the pipeline that is modified. And of course vtks or permissive license BSD and python wrapped and a very range JavaScript with them. There are some limitations that there is no transformation storage in the Decay. So that's one of the reasons why we use and IDK five four months. And further limitations that it's CPU only and all the data is to live in memory. So that was all that one wanted to show. Before the demo so here this is three slicers interface and I just created the same place Brian transform using these control points.

Andras Lasso: And in the transforms module, you can see that transform it is a thin based Brian computed from 15 landmarks and like that the transform from the parent in the transform tree is this simply in spine and the inverse of that is computed by inverting for example, if I want to apply this transfer to an image that I just write we can choose that this image is under this transform. So if I modify the transform data automatically updates the underlying image and this is what I said that it is.

Andras Lasso: Like laser resampling that it doesn't resample the full volume just if I have a slice view for this three slices and I modified the transform then that only recomputes these three slices so it's very fast. Considering that it's a highly non-linear transform with relatively large number of contractor points for a protesting Place fine. We can also.

05:15:00

Andras Lasso: apply this for four different other data types of freedom for a segmentation if I want to apply it for that Then you can see that this is internally a liberal maps of Arabia map image as well. it is a pride automatically.

Andras Lasso: Then what I also have a model surface and for this as well if I modify. the transform then it changes everything and you can see that or for example to warp this mesh the transform has to be inverted on the fly. so actually to be able to visualize this to warp a mesh and warp an image with simultaneously update the transform and it's inverse and we apply it both the surface mesh and to the image at the same time. so it is saying that and

Andras Lasso: In other presentation, it was mentioned that we would like to implement working transfers in the future. This is available for about 15 years and it's well tested tons of small things came up during these years that and special cases or performance issues that were free. so there's a lot that a lot of work went into that aware over the years. and some other useful features is somebody computer transforming any and environment and would like to make sense of that Transformer verified that transform that it can be very useful to visualize the super example, we can transform visualization and there are many different ways how to

Andras Lasso: how to visualize this transforms so

Andras Lasso: you can show 3D.

Andras Lasso: and then to the glyphs Greer's and each visualization option has many additional parameters that can be So this can happen a lot in understanding issue. That's okay why my transform is not accurate or what's like we have a visual assessment over about the equality of a displacement field so you can immediately see with this visualization if the displacement field this kind of wavy warped or that is not always obvious from the image content.

Andras Lasso: and compose the transforms in your time. So for number five go to the Transformer. He had creating you a few transforms. that I can composite them like This so there's a linear Transformer when I'm within that there's another non-linear transport, then the transform within that and then we can edit that. I created this area of Transformers and then I can even create composite transforms this way if I say that.

Andras Lasso: There is this linear transform.

Matt McCormick: Thank you. Those very nice.

Andras Lasso: And there is a transform below that I can make them a composite transform.

Matt McCormick: Yes. Yes we have. Yes.

Andras Lasso: and then

Andras Lasso: if I come here then these are two transformed transform one is the same place Brian transform to is linear and if I want to split it again into separate Transformers and just click it and then it split the Transformers again. and you can

05:20:00

Andras Lasso: also edit the transform using widgets proper simple. linear transform

Andras Lasso: I can. show an interaction and

Andras Lasso: modify for them for this Kumar has if I apply this.

Andras Lasso: So the translation rotation in playing out of plane or this way. These are just used for user interface. If you want to I guess for Transformer slicer at a lot of tools for modifying Transformers visualizing transforms and loads and save them from IDK. I formats.

Andras Lasso: This is what I wanted to show. I don't know if there's any question.

Andras Lasso: I cannot.

Matt McCormick: The metadata to Jason handle that's much nicer than it's an hdf5.

Andras Lasso: Yeah, so…

Andras Lasso: then it's an IDK H5 transforms and…

Matt McCormick: but in terms of the types of data,…

Matt McCormick: it's float and double

Andras Lasso: I come back. Let's say in the hierarchy I create.

Matt McCormick: For how the parameters are stored so 32-bit floater 64-bit float

Andras Lasso: in our transform and within that

Matt McCormick: I don't know more type supported Nice, we don't have explicitly of that constraint. Do we and The homies are proposal.

Andras Lasso: I use us Korea.

Davis Bennett: I think omeys doesn't talk about. the types of the arrays I think it just defers to the Tsar spec and the two the floating double those are supported in rb3, but I wasn't sure if you Needed a string type or anything like that. But if you can put that in the attributes. One thing that's always kind of bugged me about Czar is that we're storing numerical types in the arrays, but it's essentially, Json doesn't really do numerical types. And that's often we have attributes or metadata that describe. I mean the classic the simplest example is the fill value attribute, which should be the same type as the numerical type in the array,…

Andras Lasso: Okay, so I will say and we can and…

Davis Bennett: but because it's Json you can't guarantee that at all.

Andras Lasso: visual studio code can display the agenda 5 content.

Davis Bennett: So I've often wondered if It's the attributes where itself an extension point and you could store your attributes in something with a slightly more robust type system than just Json. It'd be a different format, but something could be nice.

Andras Lasso: Okay.

Matt McCormick: Andrases has an I threw those as text format related to that. I'm just going to speak to it too. But one of the things we did in itk for this text format and things like Jason is using specific libraries. So you'd get exact round trip conversion for the Precision for doubles which is non-trivial going from decimal to binary. And I know more and more libraries support that with the Json parches to have but it's my guarantee.

Andras Lasso: This is the IDK text format. So this is what is in the four months. So there's a composite transform and the same place Prime and a fine transform. the same content should be in the initiative file so Matt if you can tell me What to install…

Matt McCormick: …

Andras Lasso: then we can have a look.

Matt McCormick: if you go to the extensions for vs code. And then search for hcf5.

Matt McCormick: And this would be good. Yeah, I think the third one I think.

Andras Lasso: Which one the first one?

Andras Lasso: Is five web.

Matt McCormick: Yep.

Matt McCormick: and then

Davis Bennett: Sort button has a logo.

Matt McCormick: And then just open the htfi file you would be opening a source code file. And then we'll be able to look at it.

Andras Lasso: You need I guess there is money.

05:25:00

Matt McCormick: it'll be easier once we visualize it but the davis' point kind of to this you have the converse of storing doubles in. Json is not ideal, but Through the other metadata. It's stored in a non-ide deal way to some degree in strings in hdf5. So. It des. What is the transformerization? in the dimensions in a stream but we'll see that and then just the parameters in a simple 64 bit float array

Andras Lasso: next time looks like okay.

Davis Bennett: so this might be a precursor to the discussion in the later days, but I kind of wonder if we're going to human readable format for storing this stuff. I understand that linearizing an affine Matrix down to a list of numbers is dense and efficient, but it's not human readable and I would suggest it's actually a source of Errors unless the spec is very well written. So I wonder if we should think about using Json a little bit more aggressively here and defining ructure. In the metadata instead of I mean, I was kind of my last comment on the Oracle on the transforms pull request was to maybe suggest moving away from trying to serialize a matrix and instead try to serialize a transformation.

Davis Bennett: Because the transform I saw there was just a big list of coefficients which of course is essential information, but I wonder if it could be decorated with some structure that would convey to a reader or a person trying to do an implementation or make sense of a transformation. What those numbers actually mean?

John Bogovic: And yeah, so my opinion which then eventually, I'll continue and then back until I think that has Merit for sure and I think we should think about in the future. and I guess the reason that I had been sticking with this for the moment is that

John Bogovic: consuming software essentially takes the flat Matrix today or the sort of 2D array today and so adding doing the more structure would be awesome for human readability and probably for ease of interpretation, but it would also add a layer of code that would essentially have to be vert nice they structured human readable thing to the code that has to consume the thing eventually so I guess To summarize. My opinion is to stick with what we've got now and then consider that for the future. Yeah.

Matt McCormick: I think If a person who could see a human readable format and go, that represents the rotational along the coronal axis. But I wanted a translation along the criminal axis. Then that means someone with a deep level of subject knowledge. Is looking at the transform as it is serialized. Which means you've already lost?

Matt McCormick: the odds of someone.

Davis Bennett: I mean, I think you're underestimating how much? Knowledge people have who work with file formats?

Matt McCormick: No, I'm not saying the knowledge with the file formats.

Andras Lasso: she's

Matt McCormick: I'm saying that the person who knows that it should be a coronal rotation because they just rotated someone coronally. Is not going to be a file format expert.

Davis Bennett: Right somebody like me who works with data and is told it should be a rotation or it should be a translation. I want to be able to quickly debug that. so I've seen these threads where people struggle to figure out the correct order. Or interpretation of these list of numbers. And it's like come up many many times. I've seen where an affine transform expressed as a list of numbers is ambiguous. And it's easy.

Matt McCormick:

Davis Bennett: If you have one program that interprets it exactly one way but as soon as you have to cross ecosystems. Then it becomes ambiguous. So then I just ask how expensive is it to make it more explicit. Because json's cheap we have computers are really fast so we can probably to use what Json offers and throw some structure in

05:30:00

Andras Lasso: not for displacement field, but for all the or the other times it's probably text format is fine.

Matt McCormick: I see where you're coming from. But I think that's why I gravitated Using a library to read and write these which is essentially what you said of one per one program reading and writing them. Because it's just so hard to the collect basically an orientation Matrix to not get transposed. Immediately. that

Davis Bennett: Yeah, so the 100% that is a hundred percent the issue. I want to solve I think as long as we insist on storing flat arrays of numbers it will persist. so I would ask how can we use Json to solve that problem?

Andras Lasso: Yeah.

Davis Bennett: I'm making this stuff on a big.

Matt McCormick: I think there's people in your mailing lists are fully capable of transposing Json. And I wish that this was not the world we were in. but This might be a case. Where the problem. Is that you just are too good at this? But I fundamentally. Will only get a matrix transpose, two thirds of the time at best and no amount of hand holding and Detail in the Json is going to get my performance there better. I just have to write it and then test the hell out of it. and I'll

Davis Bennett: So The Proposal I made was that we don't at least for an affine Matrix. We don't represent it as a matrix. We represented as a linear transformation. parameterized by axis names and so it's super explicit. It says access with this name. the output space is a result of multiplying the axes with this name in this thing. So there's no matrix. It's Json objects.

Matt McCormick: I think that would get me to from 60% success rate. Which is but I think I might be a me problem.

Matt McCormick: be interesting to create an implementation of that and see if it's works with something like

Matt McCormick: what? Luca presented

John Bogovic: Ask why don't you go first Luca and then all I have a question for andras again at some point. but once you go first

Luca Marconato: Okay, thank you. Great presentation, sir all the day. I'm really able to be here. I'm definitely a beginner. I want to say not an expert and a beginner in a no linear Transformations. It was great to hear all of us. So what is my comment by listening to this talk is that when I presented before the three levels of possible collaboration and the first one adding some classes for nzf Transformations. The second is the focusing on ergonomics from a python perspective and the third one on transform. I am 100% convinced that the transform part is Super Soul in excellent Ways by different tools and so as a user what I would like to have them is that we have a front end library with implements with transform operation on start and by standard python objects like x-ray object or Japan's data frame, so

Luca Marconato: Is for us our data and Dr. Data and then I became I can say I want to use itk I want to use a different tool and then I get the result with the same data type. So I think that this part can be done with one or more of your tools and I still believe that before two components are something that are not available nowadays. So as a community that works it only in cff, I think that there is the need for a up-to-date Community but by the community a set of classes that mirrors the nzf class season that are validated and…

Andras Lasso: so that like to separate

Matt McCormick: Let's do that.

Davis Bennett: I mean

Luca Marconato: I also believe that somebody like the pros that we implemented could be the remote for the user.

05:35:00

Matt McCormick: Yeah, we kind of derailed on us.

Matt McCormick: So on this please go back to.

Andras Lasso: I just want to clarify that the proposal is to separate the transformation implementation itself from the rest of the infrastructure.

Luca Marconato: I have my x-ray object. I want to do separations. I want to settle transformation inside as metadata and I want to get it out and when I get it out, I want to put in another object and then the trans Delegated to other Library.

Andras Lasso: Is that… what you are describing?

Andras Lasso:

Luca Marconato: So I don't know

Luca Marconato: what you think about this. I think that other coupling between the file format and the libraries to peritomophile format. I play this objects in memories should be probably the couple by the actual implementation that the transforms the object.

Andras Lasso: Yeah, yeah. yeah, and we kind of do this with it again and vtk right now because vtk has this transformation pipeline, which is useful for real-time visualization while itk has some other useful features. For example, the composite transform writing. so we use IDK for dealing with the data IO while the transfer actually within Sizer is computed using the TKA classes so it can

Andras Lasso: It didn't lead to any problems It's kind of working nicely in this one specific example, and it should work well with the five formats and other transformation implementations, it could make sense to standardize on things that okay for a simple plan. Which Kerner do we use or details like this if someone influence they algorithm done? It is clear This is the reference implementation is in this itk class use with this parameters. And then if anybody wants to imprint in a different environment different programming language than it. It can be compared to this reference implementation.

Luca Marconato: I think that in addition set us to the separation that you implemented them. You have implemented in a vtk. And you mentioned also IDK. I think that what will be probably useful for the user is one. I can see the sterilization to a language agnostic format. That doesn't require a software to be read. So yeah, if a Specs as we discussed before could be the solution but the second as a user around I used to work with numpyre now x-ray presentation that is back from using dasca. I think it's the best way to work with us the images. So yeah, they're always this objects in memory. And what I would like to do is to say I have a single function that applied to this object given a transformation and coded the FFF gives me the result in the same object, so I don't have to

Davis Bennett: A point on that specifically about x-ray so basically an x-ray terms…

Luca Marconato: Load another Library manually convert the object is supported from valid Library already serialized to file and…

Davis Bennett: what you would want.

Luca Marconato: say now we read this file with IDK em reader and…

Andras Lasso: Or your I don't hear anymore. I think I needed.

Davis Bennett: I'm sorry. I'm here myself.

Luca Marconato: then I applied the transformation.

Davis Bennett: What you'd want is a multi-dimensional coordinate variable associated with your data that expresses the result of the data transformed with some transformation and…

Luca Marconato: So I will really say I work with x-rays and I want to get an x-ray out and similarly when I work with the geometries.

Davis Bennett: it should be evaluated lazily on a trunk by chunk basis an x-ray cannot do this today x-ray…

Luca Marconato: I work with a few hands object. There is also special Panda. I think there are the only two options probably is upon.

Davis Bennett: if you're trying to find a multidimensional coordinate that's based on a desk or…

Luca Marconato: This is the most common because it's based on shapely and integrated with the partners equal system.

Davis Bennett: a x-ray is going to put it on memory immediately.

Luca Marconato: So yeah,…

Davis Bennett: So there's an open issue about this.

Luca Marconato: let's open this object and I had a single function that gives me a transformation…

Davis Bennett: Within the Geo Community there's a sort of a division between people…

Luca Marconato: because otherwise the area be forcation of approaches. I can say I have to manually pass Upon Us object a fine Transformer or…

Davis Bennett: who use the X-ray net CDF data model where the coordinate data is rasterized and…

05:40:00

Luca Marconato: no linear extension…

Davis Bennett: then there are people…

Luca Marconato: if implemented…

Davis Bennett: who do a lot with affine transforms and…

Luca Marconato: what Joe has expects.

Davis Bennett: they would use more geotives and…

Luca Marconato: So you need to pass manually the transformation from zfora on the other side and…

Davis Bennett: these are kind of two different flavors of Geo and…

Luca Marconato: you take to find us object create another presentation.

Davis Bennett: climate data analysis. And there's a post recently on it's kind of like x-ray Forum about…

Luca Marconato: That is recognized by the Decay or other libraries and then get the result and…

Davis Bennett: what needs to change and x-ray for it to be good for supporting the transforms and…

Luca Marconato: put it back to Japan. So it still think that all the components are here,…

Davis Bennett: this of course is with a climate Focus.

Luca Marconato: but the breeds that puts them in an ergonomic way can be refined.

Davis Bennett: So they wanted to support this other community of climate scientists. But if they solve that and I think it's on their radar then they would have basically solved having coordinate that dispatches to the evaluation of say an affine transform to pull out the coordinate data and that would be generalizable to the bioimaging case where we have sort of more expressive transforms. So that's relatively recent that's the last few months. I can send you the discussion where they're talking about that.

Luca Marconato: Yeah, thanks a lot sounds really.

Andras Lasso: I think difficult it could be If you can't agree in a format how an image other data structures that represented in memory, then we cannot really go beyond and kind of have In presentation that automatically chose them based on what is the data types? I think first we would need to agree on for example how we represent an image and figure out details about how to describe access metadata if direction is part of the image or there are many small question. I think and that are still not decided and before that happens, I don't know how we could build this infrastructure that allows working on these data sets.

Andras Lasso: So, for example, we could have a adapter player that uses VT carrier IDK transforming infrastructure. We're consummate about them. We would need to know. how we represent an image how we represent the mesh and then we implemented or accordingly.

Davis Bennett: As a pandemic point but technically all medical imaging has a sample that can be resolved to latitude longitude and elevation. So we do all our Imaging on Earth and I I don't know what the geospatial people do about satellites and stuff in Mars, but I actually have a lot of hope that we can take inspiration from there tools. I mean x-ray has been huge for us and I think more broadly they have a lot of money and they've been working on similar problems or even harder problems like ocean sensors that transmit data once a week and move all around the ocean. our Imaging is a lot more tame compared to that kind of sensing. So yeah. My hope is that we can learn from them and use some of their tools.

John Bogovic: Can I back it up and ask some I guess I have two questions and who wants a question first Thomas or andras or I'm gonna choose first. I'll go in order because No, I'm gonna change my mind. I guess first andras you had mentioned something really cool,…

Andras Lasso: Okay.

John Bogovic: which was how your on the Fly can estimating the inverse of the thin plates blind that you Computing because it's necessary to transform the mesh,…

Andras Lasso: Yeah.

John Bogovic: which was super cool. And if you wouldn't mind giving some more details about that specifically, I'm curious how you're representing the inverse of the thin plates blind.

Andras Lasso: yeah, that's Very easy with the Boolean flag. That is inverse. I didn't show that I didn't actually show the inversion but it's very impressively simple So it's just a flag.

John Bogovic: Okay, so that's

Andras Lasso: That it's an invert. I only can see my screen now if I want to in there the transform then I just pick the invert button and you can see that okay transform from parent. Is this transform? And the other is computed from emerging and…

05:45:00

John Bogovic: yeah.

Andras Lasso: if I click invert then this jumps here. So this is the Transformer this is computed. So inversion is just a flag that This is inverted.

John Bogovic: Gotcha, that's helpful. So then an if I can ask just a small implementation question then is what happens when you're asking please transform, this vertex through this inverse template spline. I guess it knows that it has to do that and then does the iterative routine per vertex when it's asked for that. Is that accurate?

Andras Lasso: Yeah, yeah,…

Andras Lasso: it's decided and…

John Bogovic: Okay, great.

John Bogovic: Okay, great.

Andras Lasso: and yeah,…

John Bogovic: Then

Andras Lasso: the implantation is quite simple over us. so they're not a lot of functions that the transform has to provide. I think it's just the derivative and then the optimization is the same for all the transform. So when you implement the transform then you don't need to worry about how the inverse will be computed as long as you implement these few the classes there. in these are the only Functions that you need to implement so for our transform derivative. These are the two methods that you need to provide for a transform and everything else is implemented automatically.

John Bogovic: Nice, And the reason I was asking was if one were to serialize if that inverse in place button was written to disk essentially we'd see that text file or the hdfi file and it would have the Boolean flag in there which would mean IDK knows that yeah, okay.

Andras Lasso: yeah, yeah and Yeah, this is actually a little bit of like that. that we do that I think he doesn't have the flag. So we added the inverse in the transform type name. So then it's whatever I find underscore inverse and then we know okay, we read that and then we populate the inverse flag from that. But essentially it's a booleaf stored as a string paint.

John Bogovic: And then the cleverness is on the flying version, that's very cool. Thanks.

Andras Lasso: Yeah, yeah, and that's also very useful just to That they will go implemented or they send in in vitic and he brought it already smart and quick way. So if anybody needs anything like this, then it can be taken from vdk and then being prevented in any and any other place and the whole thing that This API is sufficient just the forward transform and derivative nothing else and also this smooth extra operation

and all these little details that Probably took a lot of time for David to figure out but now and anybody Can use that.

John Bogovic: Yeah, And then could this is a quick weird question for Thomas because you had mentioned I think the quote Nerf type nonsense as a representation of a transform. Do you think there's any benefit today of trying to represent something like that directly or is it plenty good enough to just render it to a displacement field and write that and say that's good enough for this.

Andras Lasso: Who is this question is for?

John Bogovic: for a Thomas right.

Andras Lasso: the Hastings Hastings talked before me

John Bogovic: Yeah, yes.

John Bogovic: Yeah. Yeah You had mentioned this Nerf type nonsense as a thing people do sometimes yeah.

John Bogovic: cool so meaning for the people who do it today if someone were to do this, right and say my transformation is a Nerf type thing. And wanted to share it in an interoperable way. We should just recommend please write a displacement field that represents that thing and that should be what we've recommend. Yeah.

05:50:00

John Bogovic: Yeah. Yeah good.

John Bogovic: wonderful

Davis Bennett: private question about saving displacement fields how do people feel about a ray of vectors? Or array with an extra Dimension that has length equal to the dimensionality of the data.

John Bogovic: Personally, I don't have a strong opinion. There's semantically almost the same to me and the memory light like the bites that I write to memory for those two are probably going to be identical. So I don't super care. I'll say that in image live to the library of choice. the displacement field is represented semantically as a array of vectors. But this is always red from a 4D array that we convert to that array of vectors.

Andras Lasso: In it again vitic. I think it's always like x y z in components in memory not XXX and why? so if there is no specific reason to change it, then the XYZ would be slightly preferable for compatibility.

Davis Bennett: But does idk have a pixel type that is a vector?

Matt McCormick: Unfortunately, yes.

Davis Bennett: Otherwise unfortunate it seems nice. It's an array of vectors.

Matt McCormick: Yeah, it's good. I actually expressed no negative opinion about that because Matt's cleared at me.

Matt McCormick: Hey glad too hard. I'll see a little glitter gentler and

Davis Bennett: But then there's some signal there what's the emotional reaction? why does it feel unfortunate until you see someone glaring at you? is it hard to work with?

Matt McCormick: Because I C K has both 40 arrays and raise a vectors. And isn't sufficiently opinion a opinionated about which of them. And so sometimes you've got one and what the other and then you're sad. Because you have to fight with+ types.

Davis Bennett: All right. I think I'm using the array of vectors or n Plus One D array is a wedge to get into the point that color channels also should be an array of rgbs and not an array with an extra color Dimension, but these are both things that Yeah,…

Matt McCormick: This is why.

Davis Bennett: it's about things that bug.

Matt McCormick: What you're going to do here is out all the Julia programmers and then we'll be in real trouble.

Davis Bennett: This is a bone. I have to pick with the ngff spec and I think this is something worth talking about but even in the current spec we Define semantics for axes in a way that I think isn't fully baked. So we Define these types for axes and so there's space A Time type a channel type And when you write down your scaling and translation transformation, it has to match the rank of your array. And so if you have a five-dimensional array where the last Dimension is channel, you have to write down a scaling transformation. For your channel access and…

Matt McCormick: Yeah. Yeah.

Davis Bennett: a transformation for your channel access. It doesn't mean anything and there's nothing in the spec that explains. What the difference is between a channel access and a space axis with respect to spatial transforms. But my point of view is that an affine transformation doesn't make any sense for a channel access the way most people normally use it and so clarifying some of the semantics on axes would be a big win for me. The ultimate wind would be for people to stop Channels as an extra dimension in the array but that's not going to happen. That's conventional. So the next thing we could do is be a bit more serious about defining different types of vaccines like explaining what the types mean in giving them semantics and there's some open issues about that.

Luca Marconato: So maybe a quick comment on this. We also experience this when we implemented before set of classes to their realize that we could have glasses that simply don't have information about the channel and they still work. So when we write to nzf to be compliant, we try to limit the data as expected by the specification, but then when we read it in memory with us keep the access that are actually containing a transformation. So if we read the scale that contains a Super X and one for that and one for C, we just keep the Super X

05:55:00

Matt McCormick: and it's a little bit worse than that for displacement Fields because they have to rotate tensors.

Matt McCormick: but The general point is correct.

Andras Lasso: Yeah, that's an important point that you need to specify additional dimensions are spatial or color or other because for spatial you need to apply the transformation itself. So if you rotate the whole displacement field,…

Matt McCormick: And It's worse than…

Andras Lasso: then you may need to rotate the vectors.

Matt McCormick: There's also covariant first contravant Vector fields.

Andras Lasso: That's why it depends on what the reason Of rotation,…

Andras Lasso:

Matt McCormick: And at some point you just cry and…

Matt McCormick: go back to 40.

Andras Lasso: but for example, when we change from areas to RPS coordinate system representation,…

Matt McCormick: I don't know. This is why I like having a library that I carry around that's 500 megabytes instead of implementing into this myself to be honest.

Andras Lasso: it's not enough to flip the image header,…

Andras Lasso: but we also need to flip all the vectors that are stored in the image. So

Matt McCormick: But I take back…

Matt McCormick: what I said about itk. They're being troubled that they have a vector data type. That's correct. And the 40 thing is wrong.

Andras Lasso: yeah, yeah.

Matt McCormick: These are all very good conversations. I would say.

Matt McCormick: keep them for them being productive and to writing them in formal comments you start now or it's probably tomorrow a little more energy after all the very good dissemination we had today. I think you're looking at the current spec.

Matt McCormick: and writing up formal comments tomorrow Is a good idea?

Matt McCormick: so I think it'd be good for folks to what are some other conclusions that people have from today on what they'd like to do going forward?

Luca Marconato: Yeah, so I think continue tomorrow is with fresh energy. It's gonna be probably more productive and also quite tidier in Europe. one of the goals I think with probably everybody would agree is to simplify the interface to events.

Matt McCormick: Yes.

Luca Marconato: Yes transformation. So now if you use a woman's are buying we read I tell you a few months ago, but we could read only the 0.4 without the new current Transformations. And so that could be a way to goals in the direction of itk and you don't have to open the files yourself. You can use some Python apis and you can read this with the apis and then this could be a use by other and libraries to be locked up.

Luca Marconato: And for instance having a validator could be part also the task.

Matt McCormick: I'm curious, the validators another question that came up was. an ngff transformation created with big Warp can it be red with? spatial data

Matt McCormick: and tried that

Luca Marconato: Let's go. we could try definitely these things so in a special data, we don't support the linear Transformations. So all that part will not work. I can't imagine that the final transformation should work if they don't work. It's probably some Tiny Box that appears as because we never really tried Crossings between different tools that implemented a specification because specification unfortunately still not fully agreed on but I believe that the Samantha of Alpha hour or some little debugging and should work.

06:00:00

Luca Marconato: Just to give an example and we use izer pi to write the transformation to disk but then the transformation and the 0.4. So we also have a callback that we use after to adjust the transformation. then 0.5. So to say the problem is that almost are by when reading the data expects the transformation with 0.4. So initially we try to just keep the 0.5. This was leading to an error in user Pi. So we had a few options to deal with this one was a let's fix on their pie. The second is let's keep both the actions in the files. This is neither near 0.55. What is a quick fix until the transformation are fully accepted and almost are by accept only the 0.5. So we went for this this means that what we write is by Design not the 0.5 is this weird mix that we are absolutely interested in addressing and it's also one of the reasons why if there is a common tool that does

Luca Marconato: so almost there bike and build on top we can build on top. We don't have to worry about this edge cases that break the io otherwise

Davis Bennett: So something that you could use today would be identities are where I've defined identic models for the basic Bohemian GFF classes. And you could make a 0.5 subclass. That has all the semantics you want and this in between 0.4 and 0.5. I was tracking the latest branch of omgff until things slowed down. So I don't think the transforms are in there, but it would be as easy to implement them there as it would be anywhere else and then you would just have your own sleep like all it does is Right Json, So it would do that and it would validate it because it's just paidantic.

Luca Marconato: yeah, I think that adding this, as a recognized the community resource that everybody is aware of and are interested in using and contribute to will be the way to go and the reason why we didn't complete the io to be fully compatible with 0.5 is that we were waiting until

Luca Marconato: Are a source of reference rubeners? So when the transformation specification is mass domain and the new comes if that direction is officially released I expected in a short amount of time.

Everybody will convert to our ones tools we included so we just drop out the code that we've wrote that as a temporary work around and just it's So I think to stay practical and during this octo maybe we could all start from the code that you just mentioned and start from there and build the readers and writers who are in cff or take the code and maps with some existing code but adding a place that we can finding it up. We all agree on.

Davis Bennett: Yeah, I mean this is something that I flagged years ago that Omi zarpai is technically deficient because it doesn't expose models of the data described in the spec. So umizarpai works procedurally and there is no class you could import that represents the data structures defined in the spec. So it's very hard to extend or work with and that's a technical deficiency. That should be fixed by those developers. I mean, that's why I did identics are because all means our pie wasn't sufficient for my purposes.

Luca Marconato: I agree working with that was not straightforward because of the reason you mention.

Andras Lasso: And that's why a Json base format or service describing human readable format can help a lot because then you don't need to write a library you use the text editor to define the file format and So that's not the official format. It would be still nice to have a human resource presentation for a transform that anybody can edit your tax credit or create your text rate or if you want to add some attributes new type anything just with X8 or I can create the transform and I don't need to write any code. so it could be used as a communication tool and debugging tool so it would be nice to have a figure out the way how to get there to have a Decent base or we were invisible. five format for storing or this information

Matt McCormick: I think it would be ideal to start from the spec as suggested but I think hopefully start from something that is language agnostic in the schema or what's been proposed in the schema because then we can work with python, but we can also have that implements in other languages and also avoid issues where

06:05:00

Matt McCormick: yeah, this idea of there's bugs between the implementation and the spec which is come up. in the past two so if it's possible to

Matt McCormick: start from the schema. And build out a working implementation there. that I think to the same idea help us get to a usable useful product that has a stack of associated with it.

Matt McCormick: so I know John, he's updated the recent versions and there were Jason schemas. associated with that another option that Josh Moore who has suggested that I've looked to two Is using a link at Mel. I don't know if there's any amount of familiar with that. but we could look at starting with that and…

Luca Marconato: Yes.

Matt McCormick: then generating the

Matt McCormick: all the bread code

Luca Marconato: Uncommon there so I mean mostly a lot of work done by Johnny's Saunders and Ryan Lee, but they're actually working on an array specification. I'm actually working. There only x-ray part but I think it will be great to get in touch with them as well because what we were mentioning here of actually

the serialization with different languages and also different types of race. That people work with they're exactly going in that direction So yeah, that will be quite cool. It's not entirely there yet for ready to multidimensional array and started simple.

Luca Marconato: but yet there's work from that site from Lincoln Mel on really getting these erased specifications and Maybe to come back to an earlier point there as well also with for example with the fine transform. I think it was the point that Dave is made that can we have it a little bit more structured? Also, their Death part of supporting labeled arrays in that sense would go in that direction where I mean ultimately with the affine transform you can also say okay, it's labeled array where this particular part of the affine transform is corresponding to this. So it goes in that direction as well.

Matt McCormick: Are they on zulip or…

Luca Marconato: I can make

Matt McCormick: could you put a link to their work on zulip or invite them up?

Luca Marconato: Yeah, and…

Matt McCormick: It's been more energy.

Luca Marconato: I can and I can also maybe link to a link email Ray chat and they're also on skypaths. I can maybe see if they can actually join this feed as well. I can get in touch with them. Yeah.

Davis Bennett: Because something that I think would be a good use of effort and hackathon to write code in whatever languages convenient but parses and validates the Json documents that the spec will produce.

Davis Bennett: In a way that could be reused from a consumer…

Matt McCormick: of

Davis Bennett: who just wants to work with the metadata but also as a way of kind of smoke testing certain aspects the semantics of the spec I think this has not been done for previous versions of Omega GFF. And so I have found a lot of weird things when I actually wrote code that tries to implement the semantics to find in human language in the spec, but would never had not yet been written down as a validatable basically program. So I think the spec will be made much stronger if there's a tight Loop between the human language and the Machine language and they can, inform each other.

Matt McCormick: I mean at some point just got to Write code that serializes and IDK, I find transform into this. That's the absolute minimum.

06:10:00

Matt McCormick: That sounds like the sort of thing to do.

Andras Lasso: And it would already bring up a lot of interesting questions. if you want to save the file transform, there are some additional parameters for example Center of rotation and I take a transform where would be put that in a way that survives? at least some manipulations it would be like even just a

client transform saving and loading would be an accomplishment but if you can a few of the other transforms or have an idea how to handle or the four transform types and then yeah, it would be great.

Bradley Lowekamp (NIH/NIAID Contractor): there was the General comment made earlier with IDK transforms is that many of them can be boiled down to affine homogeneous coordinates with those translation transforms and that a lot of the differences are due to the iPhone transform being parameterized differently for registration, and You're bringing up the center of transform is one of those parameters that's really optimized or that really designed for the registration optimization. And I would think that is something that would just go away when you just kind of save it as a regular affine transform. It's translation.

Andras Lasso: It could go away but it would be much nicer because if you could keep it so presentation inside to reuse the center of rotation in the user interface as well. So if you have a widget that modifies the transformed along that point we can save the center of rotation. So if the homies are format cannot store metadata like this, then we will still need to use IDK area 5 to say the data without any data so

Andras Lasso: if we modify the transform and then the metadata doesn't survive or certain part of the metadata to survive, that would be But at least to have an option some way to store this additional metadata would be needed to be able to offer it as a full replacement of the current file format.

Davis Bennett: I personally would like a center of rotation represented in the metadata. I think that's how most people think about transforming their images. So if it's available in our metadata budget, I would strongly Advocate putting that in there.

Andras Lasso: And then that will be a good use case then for. storing additional metadata that is optional right doesn't really need it.

Matt McCormick: It's not optional.

Andras Lasso: It's different. Yeah.

Matt McCormick: Once someone puts that in there. I mean, I agree that we need Center of rotation, but we can't describe it as optional. Because if we do half of the people will write it and half of the people won't be able to read it and the whole thing will fall apart.

John Bogovic: Yeah, I have opinions, but we should talk tomorrow about this in particular.

Matt McCormick: Yeah, yeah. Yes, so that's a good topic for tomorrow. And I think we're near the end and it is very late in Europe for our friends who have probably stayed with us. And this was extremely productive and informative good learning experience for me and thank you for everyone and participated today. I suggest that we conclude the day. And pick up the torch again after some rest. It is Friday, too. So yeah, appreciate everyone being here on a Friday. Wherever you are, so let's conclude for the day and continue again tomorrow. John I'll let you

Matt McCormick: Share with people tomorrow again. We'll meet on the virtual Island it is on Earth, we'll call it Earth. It's not latitude and longitude but there's a URL that's on the event website and in the email I sent out to everyone to registered beforehand. So let's meet there and thank you again to everyone who came today and participated.

Matt McCormick: happy hacking for the rest of the day or happy sleeping and we'll see you again all tomorrow.

06:15:00

Luca Marconato: Thanks for organizing ultimate. Okay.

Matt McCormick: You bet. Thanks for coming.

Matt McCormick: That's

Meeting ended after 06:15:53 👋