

Visualization use cases

At the March 2018 DM-SE mini-JTM, there was a session "DM/SE visualization discussion for commissioning". As it turned out, an hour of the session ranged rather widely across visualization use cases across the whole of development, commissioning, and early operations.

Below is a list of the use cases, transcribed from the whiteboards in the meeting room. In the coming days the intent is to expand on these use cases by adding information that clarifies the use cases, describes whether they are pure visualization requirements or involve LSST-specific analytics, and how these might be - or already are - addressed by the Science Platform SUIT components or by other libraries or systems (e.g., `matplotlib`, `Bokeh`).

Priorities in the last two columns are keyed as follows: 1 – Critical, 2 – Nice to have, 3 – Optional.

No.	Use case key topic (from original whiteboard)	Initial clarification	Commissioning Priority	DM Priority
1	Visualization of all-sky "stuff" (a/k/a statistics, e.g., MAF metrics)	This is the display of quantities that are evaluated across the entire (LSST-observed) sky and are desirable to be able to explore at a range of scales from all-sky down to the finest-grained level at which they are defined. KB: This becomes more important with LSSTCam and particularly the SV surveys when we start having larger contiguous regions of sky coverage.	KB: 1 BS: 2	LPG: 1
2	Collections of thumbnails	Meant to be around a designated set of objects or sources (if sources, this could be a time-dependent selection of apparitions of the same object).	KB: 1 BS: 3	LPG: 2
3	Brushing and linking between scatter plots and thumbnails	Could mean: use brushing and linking to select the sources around which to request thumbnails, or: use brushing and linking to narrow the displayed set of thumbnails from an already-requested set.	KB: 2 CFC: 1 BS: 1	LPG: 2
4	Extensible displays	Because the set of interesting specific visualizations far exceeds what could be provided centrally, make it possible for users to define visualizations, constructed from the lower-level visualization <i>capabilities</i> in the system (or an external library), in a way that makes them straightforward to apply on demand.	CFC: 3 BS: 2 ("widgets for operator")	LPG: 3 (Can be done in a notebook by defining new projections of the data)
5	Turn on and off detection entries in tables, plots based on flags (i.e. with checkboxes)	Applies to any LSST catalog entry with packed Boolean flag fields, including Source, Object, ... These flag selections are meant to participate in whatever more general brushing and linking environments are provided - e.g., to permit sources/objects with designated flags to be shown or hidden in, say, a color-color plot or in an overlay over an image.	CFC: 2 BS: 2 (extension of 3)	LPG: 2
6	Quick access to schema descriptions	(Gregory Dubois-Felsmann doesn't remember exactly what was meant when this was raised in the meeting; the following is a guess.) Perhaps: given a tabular query result, provide explanatory material for the table schema(s) involved, including, say, links to documentation, units, data model information such as IVOA UCDs and VO-DML data model information.	CFC: 3 BS: 3 (seems well covered already, would be nice to have hyperlinks to each schema entry)	LPG: 3 (I can get at the schema in other ways - this will not hinder DM science validation)
7	"Standard" image interactions in a responsive way: scale, stretch, zoom, pan, colormap and basic computation and image analysis capabilities.	There was a general desire for stretch/scale changes to be "fast". In the meeting we discussed at least three different interpretations of color mapping: rendering of single-channel image data with a color that depends on the flux value (or whatever other variable is represented by the pixel values); rendering of single-channel image data with a hue or a partially transparent color overlay based on an additional channel; generation of color from 3 independent channels. For future purposes we'll treat <i>this</i> use case as referring only to the first of the three; the other two have their own use cases below. CFC: This needs to include interactive hierarchical capabilities - e.g. display of a full FOV image consisting of 21 science rafts + 4 corner rafts in some definable spatial compression /sampling scheme (e.g. box average, max value, median filter etc..) to make the full FOV image manageable. Interactive feature need to include selecting a specific science or corner raft for more detailed display, followed by selecting a given sensor on a raft for pixel level display and computational (IRAF imexamine like) operations. Other use cases listed here can be implemented through this one.	CFC: 1+++ KB: 1 BS: 1 (ds9+imexam functionality)	LPG: 1
8	On-demand pixel-level colorized images	Generation RGB color images based on the combination of three single-channel images (at a minimum: single-channel images with precise registration with each other, but it need not be limited to this). Both the "traditional" and the "hue-preserving" algorithms should be supported.	CFC: 2 BS: 3	LPG: 2 (Can be done in a notebook)
9	Colorized overlays on greyscale images (with tunable alpha)	This refers to the rendering of single-channel image data with a hue or a partially transparent color overlay based on an additional channel (e.g., colorizing a single-channel flux image by the per-pixel variance). Note that mask-overlay display could be treated as an instance of such a capability, but for tracking purposes we'll treat it separately in #19 below, as mask overlays benefit from specialized UIs/APIs for extracting and naming bits from packed bit masks. CFC: Do we know at this date (Sep 27, 2018) what the schema is for the data planes the accompany an image (e.g. bad pixel mask, variance, background image, etc...)?	CFC: 1 BS: 2	LPG: 1
10	Crossfading between images	Using a slider, allow the continuous variation of what is displayed based on two over-plotted images of the same region (whether in sky or x-y coordinates), moving from displaying one or the other image through displaying a superposition of the two. Alternatively, using a slider or a movable "curtain" UI element to wipe one image across the other. CFC: Nice to have but not essential for commissioning.	CFC: 2 BS: 3	LPG: 2 (Cannot do in a notebook but also not essential)

11	Graphical instrument navigation	<p>Display of mosaics of image data from the focal plane, or other data displayed in focal plane coordinates, with UI elements that make the components of the Camera assembly hierarchy (rafts, 3-CCD sets associated with a REB, CCDs, and amplifier segments) selectable. It is desired for that selection to then be usable to control other behaviors of visualizations, e.g., limiting displays of the DIASources from a visit to those detected on the selected subset of the focal plane.</p> <p>CFC: See use case number 7 with hierarchical display capabilities per notes.</p>	<p>CFC: 1</p> <p>BS: 1 (selected metrics distribution over FOV)</p>	LPG: 1
12	<p>Full focal plane visualization</p> <ul style="list-style-type: none"> Flux values? Zoom in (interactive) 	<p>The "?" represents an initial pass of the discussion in which we decided not to drill down into what it really means to display a full FP image as a whole on a screen (given the ~3 orders of magnitude of summarization required).</p> <p>As a matter of implementation, the people in the room were open to doing this as an application of HiPS. In some cases this is a bit of a creative misuse of HiPS, which is sky-coordinate-oriented, but this seems to still be a meaningful and useful operation. More discussion, and perhaps a demonstration, would be appropriate.</p> <p>(On-sky images could be converted to HiPS mapped onto their actual pointing on the sky; other images (e.g., flats) could be mapped to a pro-forma pointing in the HEALPix grid.)</p> <p>CFC: Displaying the full FOV image in the context of the surrounding sky would be very useful in commissioning - what are the neighboring bright source with respect to the displayed image? (LPG +1)</p> <p>CFC: Could be combined with use case 7.</p>	<p>KB: 1</p> <p>CFC: 1</p> <p>BS: 1</p>	LPG: 1
13	All the things in focal plane (image, variance, bad pixel masks etc...)	CFC: These need to have a toggle function for visibility and maybe variable transparency too.	<p>KB: 1</p> <p>CFC: 1</p> <p>BS: 2</p>	LPG: 1 (Also covered in #7 & #12)
14	Summarized full focal plane images	<p>Precomputed full-focal-plane images might be made available based on standard aggregation functions, e.g., mean, median, max, or stretched to show noise. Other aggregation functions should be available on-demand, including ones implemented as callbacks to user (Python) code. (Noting that a full summarization of a 3-gigapixel image down to the O(1 megapixel) images suitable for screen display may be time-consuming.)</p> <p>KB: pandas has some nice grouping and aggregation functionality that would certainly be useful. This might be more of data analysis than visualization directly, but in any case, I think we definitely want this functionality and a visualization wrapper pandas might get most of the way there.</p> <p>CFC: This is part of use case 7 per added notes.</p> <p>LPG: It is important to have several statistical aggregators available as a toggle</p>	<p>CFC: 1</p> <p>BS: 1 (extension of 7 to include localized stats)</p>	LPG: 2 (The computation of an aggregated image can be done in a notebook an the resultant image redisplayed)
15	Display of image backgrounds (pixels, models, and residuals)	Also may want to compare to other properties of catalogs and perhaps PSF models etc.	<p>CFC: 1</p> <p>BS: 2 (difficult to interpret what this would be useful for other than offline diagnostics)</p>	LPG: 1
16	Display of LSST data objects with natural on-image interpretations: (Heavy)Footprints, PSF models at a point, PSF variation models or measurements	Footprints should be clickable to facilitate investigation into deblending computations.	<p>CFC: 2</p> <p>BS: 3</p>	LPG: 2 (I can do this in a notebook)
17	Vector field plots (e.g., astrometric residuals, PSF moments or other technical information from the EFD)	<p>The scale factor must be adjustable.</p> <p>KB: matplotlib quiver plots, for example? We definitely need something along those lines for astrometric residuals and PSF characterization</p> <p>CFC: This seems like something that a visualization would be used to display, but the computation of the vector/quivers is external to the display tool.</p>	<p>CFC: 1</p> <p>KB: 1</p> <p>BS: 1</p>	LPG: 2 (I can do this in a notebook)
18	Per-source drill down	<p>Select a source/object (in a table, over plotted on an image, in an x-y plot) and be able to link to additional visualizations or data displays related to that source/object/</p> <p>KB: It would be very nice to be able to click on a source (or collection of sources) in a pixel-level image and have callback to the notebook aspect to run further analysis on those sources. Basically, brushing and linking integration of pixel-level image visualization with the notebook aspect.</p>	<p>KB: 2</p> <p>BS: 2 (thumbnails, stats of all epochs)</p>	LPG: 1
19	Individual mask planes (including: across a full focal plane; for coadds)	"Individual" apparently meaning that one or more mask planes to be displayed could be selected programmatically or by UI, with flexible assignment of colors and transparency.	<p>CFC: Isn't this covered by use case #9</p> <p>BS: 3</p>	LPG: Is this already implemented? (Gregory says yes)

20	Brushing and linking	<p>Across images, histograms, and scatterplots (on the board these pairs of those were called out: image-hist, image-scatter, hist-scatter, but hist-hist, scatter-scatter, and so on also make sense). Support pairwise scatterplots (e.g., displays of the matrix of all x-y plots derivable from a set of N variables).</p> <p>KB: Bokeh / holoviews / datashader offers much of this capability, for example</p> <p>CFC: PyVis tools does this.</p> <p>KB: Brushing and linking between matplotlib plots and python objects in a notebook is definitely important. In addition, it would be really useful to be able to examine an image (e.g., in Firefly), overlay catalog objects, select objects of interest, and then be able to access that selection from a notebook. In other words, I am asking for a linkage between pixel-level images, object catalogs, and the notebook aspect.</p>	<p>CFC: 1</p> <p>KB: 1</p> <p>BS: 1</p>	<p>LPG: 1 (Very important but can be achieved with PyVis or other similar tools in a notebook. We should think about whether we let everyone develop individual linked plots or if we should invest some effort into developing notebooks specifically for this.)</p>
21	<p>Chips and optics</p> <ul style="list-style-type: none"> residual positions relative to nominal model 	<p>E.g., display mean per-chip offsets, from an astrometric analysis, relative to the nominal or previously computed positions of the chips.</p> <p>CFC: This seems to be something tat would come out of a Python notebook - important tp do, but not intrinsic to the display tool.</p>	<p>CFC: 2</p> <p>BS: 3 (covered under 11, ok as notebook)</p>	<p>LPG: 3</p>
22	Flip-books per source	<p>Understood to be temporal in nature. Display of image cutout flip-books for selected sources. Automatic definition of default cutout dimensions based on source properties (e.g., smaller for point like sources, larger for extended sources, perhaps including the full parent footprint for deblended sources). Flip books for (raw, calibrated, etc.) flux images; for source models computed for each epoch; for residuals (e.g. from difference imaging vs. templates, or image-to-model residuals, where the model could be per-epoch or time-integrated).</p>	<p>KB: 2 (?)</p> <p>CFC: 2</p> <p>BS: 2 (definitely needed for difference imaging diagnostics)</p>	<p>LPG: 2</p>
23	Mean color residuals for single epochs projected into CCD coordinates and integrated over many stars (with or without nominal photometric corrections)	<p>This is the photometric analog of the well-known DES focal-plane astrometric residual display.</p>	<p>CFC: 2 (this seems like a notebook thing)</p> <p>BS: 1 (zeropoint distribution, covered under 11, probably ok as notebook)</p>	<p>LPG: 2 (Can do this with PyVis in a notebook)</p>
24	Instrument footprint on all sky (or all-sky) images	<p>This could mean both:</p> <ul style="list-style-type: none"> display a nominal focal plane footprint at any desired location on the sky, e.g., for use in understanding planned observations; and display over a coadd the outlines of the single-epoch images used to generate the coadd, or of all the single-epoch images that contain a given source. 	<p>CFC: 1 (see use case #12)</p> <p>BS: 1 (nominal footprint with some catalog of bright sources i.e. "finding chart")</p>	<p>LPG: 2</p> <ul style="list-style-type: none"> The first of these is implemented already Second - requires work on the table format of for the data. Moving to an obsCore interface. Functionality overlaps portal and firefly
25	Overlay R90 Petrosian ellipses on images for selected sources	<p>Or any other spatial visualizations of the source/object models. This could include display of the fitted parallax and proper motion model for a Milky Way object, with time ticks corresponding to observation epochs.</p>	<p>CFC: 3</p> <p>BS: 1 ("whisker plot", ok as notebook)</p>	<p>LPG: 2 (Can do this with PyVis in a notebook)</p>
26	<i>gri</i> offsets (sky) from coadds with drill down to single epoch <i>gri</i>	<p>Including image on chip scale, PSF models.</p> <p>CFC: Note sure what is meant here.</p>	<p>BS: 2 (thinking easy diagnostics of color term, zeropoint changes)</p>	
27	Coding of (X-Y plot or image catalog overlay) points by color, size, shape by 3rd variable, in spatial projections (CCD, sky, etc.)	<p>Really applies to any x-y plot, not just spatial plots.</p>	<p>CFC: 3 (seems like a notebook thing)</p> <p>BS: 2 (related to brushing and linking)</p>	<p>LPG: 3 (This can be a useful diagnostic tool but can be done with PyVis in a notebook). Gregory says this is already implemented.</p>
28	Plot source density maps Example: density of (DIA)Sources compared to sky brightness (e.g., colormap and contour)	<p>(further clarification of the "colormap and contour" would be helpful)</p>	<p>KB: 1</p> <p>CFC: 1</p> <p>BS: 3</p>	<p>LPG: 2 (Can do this with PyVis in a notebook)</p>
29	Pixel-level images with source models subtracted to see visualize residuals with respect to the model	<p>Could be useful for large galaxies, bright stars, evaluating deblending</p> <p>CFC: Could be useful for stray and scattered light analysis if the source model include ghost images from internal reflections of bright sources.</p>	<p>KB: 2</p> <p>CFC: 2</p> <p>BS: 2</p>	<p>LPG: 2</p>
30	Suggested pixel-level image visualization requirements	<p>Ability to overlay sources, along with their model ellipses; must be able to use multiple projections, e.g., instrument, sky; must be able to pan and zoom and adjust color scale quickly; must be able to overlay masks; must be able to see map value and coordinates at mouse location; a "magnifying glass" / zoom-in inset similar to ds9 would be nice to have</p> <p>CFC: This should be merged with use case #7. (LPG +1)</p>	<p>KB: 1</p> <p>CFC: 1</p> <p>BS: 1</p>	<p>LPG: 1</p>