



CLIJ2 cheat sheet: ImageJ macro 1

GPU-accelerated image processing in Fiji



	Operation	Parameters	Result	Dim	Examples
Basics / Wrangling	Initialize CLIJ	[], HD, GFX or CPU			<code>run("CLIJ Macro Extensions", "cl_device=[]");</code>
	Push			2D 3D	<code>// send current image to GPU input = getTitle(); Ext.CLIJ2_push(input);</code>
	Pull			2D 3D	<code>// get result image from GPU back Ext.CLIJ2_pull(output);</code>
	Create	1024, 1024, 8		2D 3D	<code>Ext.CLIJ2_create2D("new2D", w, h, bitDepth); Ext.CLIJ2_create3D("new3D", w, h, depth, bitDepth);</code>
	Convert			2D 3D	<code>Ext.CLIJ2_convertFloat(input, "result_float"); Ext.CLIJ2_convertUInt8(input, "result_uint8"); Ext.CLIJ2_convertUInt16(input, "result_uint16");</code>
	Copy				<code>// duplicate Ext.CLIJ2_copy(source, result);</code>
	Copy slice	 , 50		2D 3D	<code>// put a slice into a stack Ext.CLIJ2_copySlice(stack, slice, sliceIndex); // copy a slice out of a stack Ext.CLIJ2_copySlice(slice, stack, sliceIndex);</code>
	Crop	 , 20, 20		2D 3D	<code>// crop image Ext.CLIJ2_crop2D("original", "cropped", x, y, width, height);</code>
	Paste	 "cli", 9, 9		2D 3D	<code>// paste image Ext.CLIJ2_paste2D("cropped", "target", x, y);</code>
	Release				<code>// free / release memory occupied by an image Ext.CLIJ2_release("image name");</code>
Clear				<code>Ext.CLIJ2_clear(); // empty GPU memory</code>	
Spatial transforms	Rotate by 90 degrees			2D 3D	<code>Ext.CLIJ2_rotateClockwise(input, result);</code>
	Rotate	 , 45, true		2D 3D	<code>Ext.CLIJ2_rotate2D(input, result, angle, rotateAroundCenter);</code>
	Flip	 , true, false		2D 3D	<code>Ext.CLIJ2_flip2D(input, result, flipX, flipY); Ext.CLIJ2_flip3D(input, result, flipX, flipY, flipZ);</code>
	Translate	 , 20, 20		2D 3D	<code>Ext.CLIJ2_translate2D(input, result, shiftX, shiftY);</code>
	Affine transform	 "center rotate=45 -center"		2D 3D	<code>transf = "center scale=2 rotate=45 -center"; Ext.CLIJ2_affineTransform2D(source, result, transf);</code>
	Deform / warp			2D 3D	<code>// warp image Ext.CLIJ2_applyVectorField2D(source, vectorFieldX, vectorFieldY, result);</code>
	Projections			3D -> 2D	<code>Ext.CLIJ2_argMaximumZProjection(in, result, arg_z); Ext.CLIJ2_standardDeviationZProjection(in, result);</code>





CLIJ2 cheat sheet: ImageJ macro II

GPU-accelerated image processing in Fiji



	Operation	Parameters	Result	Dim	Examples
Filters	Gaussian blur	 , 10, 10		2D 3D	<code>Ext.CLIJ2_gaussianBlur2D(input, result, sigmaX, sigmaY);</code>
	Difference of Gaussian	 , 2, 2, 20, 20		2D 3D	<code>Ext.CLIJ2_differenceOfGaussian2D(input, result, sigma1x, sigma1y, sigma2x, sigma2y);</code>
	Invert			2D 3D	<code>Ext.CLIJ2_invert(input, result);</code>
	Laplace			2D 3D	<code>Ext.CLIJ2_laplaceBox(input, result);</code>
	Mean	 , 5, 5		2D 3D	<code>Ext.CLIJ2_mean2DBox(input, result, radiusX, radiusY);</code>
	Median	 , 5, 5		2D 3D	<code>Ext.CLIJ2_medianSliceBySliceBox(input, result, radiusX, radiusY);</code>
	Minimum	 , 5, 5		2D 3D	<code>Ext.CLIJ2_minimum2DBox(input, result, radiusX, radiusY);</code>
	Maximum	 , 5, 5		2D 3D	<code>Ext.CLIJ2_maximum3DBox(input, result, radiusX, radiusY, radiusZ);</code>
	Top-hat	 , 25, 25, 0		2D 3D	<code>Ext.CLIJ2_topHatBox(input, result, radiusX, radiusY, radiusZ);</code>
	Logarithm / Exponential			2D 3D	<code>Ext.CLIJ2_logarithm(input, result);</code> <code>Ext.CLIJ2_exponential(input, result);</code>
Segmentation / labeling	Threshold	 "Otsu", 127 or 		2D 3D	<code>Ext.CLIJ2_threshold(input, binary_result, 127);</code> <code>Ext.CLIJ2_thresholdOtsu(input, binary_result);</code> <code>Ext.CLIJ2_localThreshold(input, threshold_image, binary_result);</code>
	Mask	 		2D 3D	<code>// mask an image</code> <code>Ext.CLIJ2_mask(input, mask, result);</code>
	Connected components			2D 3D	<code>Ext.CLIJ2_connectedComponentsLabelingBox(binary_in, labelmap_out);</code>
	Label to mask	 , 4		2D 3D	<code>Ext.CLIJ2_labelToMask(labelmap_input, mask_result, label_index);</code>
	Mask labelled	  , 4		2D 3D	<code>Ext.CLIJ2_maskLabel(input, labelmap, result, label_index);</code>
	Exclude on edges			2D 3D	<code>Ext.CLIJ2_</code>
	Label spots			2D 3D	<code>Ext.CLIJ2_labelSpots()</code>
	Label Voronoi			2D 3D	<code>Ext.CLIJ2_labelVoronoiOctagon(labelled_spots, label_voronoi)</code>














































CLIJ2 cheat sheet: ImageJ macro III

GPU-accelerated image processing in Fiji



	Operation	Parameters	Result	Dim	Examples
Math	Set	 , 100		2D 3D	<code>Ext.CLIJ2_set(result, pixel_value);</code> <code>Ext.CLIJ2_setRampX(result);</code> <code>Ext.CLIJ2_setColumn(result, column_index, value);</code>
	Absolute x			2D 3D	<code>Ext.CLIJ2_absolute(input, result);</code>
	Add / Subtract	  or 50		2D 3D	<code>Ext.CLIJ2_addImages(summand1, summand2, result);</code> <code>Ext.CLIJ2_addImageAndScalar(input, result, scalar);</code> <code>Ext.CLIJ2_addImagesWeighted(in1, in2, result, a,b);</code>
	Multiply / Divide	 , 2		2D 3D	<code>Ext.CLIJ2_multiplyImages(input1, input2, result);</code> <code>Ext.CLIJ2_multiplyImageAndScalar(input, result, n);</code> <code>Ext.CLIJ2_divideImages(divident, divisor, result);</code>
	Multiply Matrix	 		2D 3D	<code>Ext.CLIJ2_multiplyMatrix(matrix1, matrix2, matrix_out);</code>
	Equal = Not Equal !=	 		2D 3D	<code>Ext.CLIJ2_equal(source1, source2, result);</code> <code>Ext.CLIJ2_notEqual(source1, source2, result);</code>
	Greater / Smaller	 		2D 3D	<code>Ext.CLIJ2_greater(source1, source2, result);</code> <code>Ext.CLIJ2_smaller(source1, source2, result);</code> <code>Ext.CLIJ2_smallerOrEqual(source1, source2, result);</code>
	Equal = Not Equal !=	 		2D 3D	<code>Ext.CLIJ2_equal(source1, source2, result);</code> <code>Ext.CLIJ2_notEqual(source1, source2, result);</code>
Binary Images	PullBinary			2D 3D	<code>Ext.CLIJ2_pullBinary(String image);</code>
	Draw line / box / sphere	10, 10, 50, 50		2D 3D	<code>Ext.CLIJ2_drawLine(img, x1, y1, z1, x2, y2, z2, thickness, value);</code> <code>Ext.CLIJ2_drawBox(img, x, y, z, width, height, depth, value);</code> <code>Ext.CLIJ2_drawSphere(img, x, y, z, r_x, r_y, r_z, value);</code> <i>// Pixels apart from the line/box/sphere are untouched!</i>
	Pull regions of interest			2D 3D	<code>Ext.CLIJ2_pullAsROI(binary_image);</code> <code>Ext.CLIJ2_pullToROIManager(binary_image);</code>
	Fill holes			2D 3D	<code>Ext.CLIJ2_binaryFillHoles(source, result)</code>
	Not			2D 3D	<code>Ext.CLIJ2_binaryNot(source, result);</code>
	And / Intersection	 		2D 3D	<code>Ext.CLIJ2_binaryAnd(operand1, operand2, result);</code> <code>Ext.CLIJ2_binaryIntersection(op1, op2, result);</code>
	Or / Union	 		2D 3D	<code>Ext.CLIJ2_binaryOr(operand1, operand2, result);</code> <code>Ext.CLIJ2_binaryUnion(operand1, operand2, result);</code>
	XOr	 		2D 3D	<code>Ext.CLIJ2_binaryXOr(operand1, operand2, result);</code>
	Dilate/ Erode			2D 3D	<code>Ext.CLIJ2_dilateSphere(source, result);</code> <code>Ext.CLIJ2_dilateBox(source, result);</code> <code>Ext.CLIJ2_erodeSphereSliceBySlice(input, result);</code>





CLIJ2 cheat sheet: ImageJ macro IV

GPU-accelerated image processing in Fiji



Operation	Parameters	Result	Dim	Examples
Vectors, arrays, matrices, graphs & meshes	Spots to point lists		2D 3D	<code>Ext.CLIJ2_spotsToPointlist(binary_spots, pointlist);</code> <code>Ext.CLIJ2_labelledSpotsToPointlist(labelled_spots, pointlist);</code>
	Generate distance matrix		2D 3D	<code>Ext.CLIJ2_generateDistanceMatrix(pointlist1, pointlist2, distance_matrix);</code>
	Generate touch matrix		2D 3D	<code>Ext.CLIJ2_generateTouchMatrix(label_map, touch_matrix);</code>
	Touch matrix to mesh		2D 3D	<code>Ext.CLIJ2_touchMatrixToMesh(pointlist, touch_matrix, mesh);</code>
	Distance matrix to mesh	, 2.5	2D 3D	<code>Ext.CLIJ2_distanceMatrixToMesh(pointlist, distance_matrix, mesh, max_distance);</code>
	Mean of touching neighbors		2D 3D	<code>Ext.CLIJ2_meanOfTouchingNeighbors(values, touch_matrix, mean_values);</code>
	Count touching neighbors		2D 3D	<code>Ext.countTouchingNeighbors(touch_matrix, count_vector)</code>

Working with arrays and tables	Statistics		2D 3D	<code>Ext.CLIJ2_statisticsOfBackgroundAndLabelledPixels(image, labelmap);</code> <code>Ext.CLIJ2_statisticsOfLabelledPixels(input, labelmap);</code>
	Push Results Table		2D 3D	<code>Ext.CLIJ2_pushResultsTable(image_name);</code>
	Push Results table column	, "Mean"	2D 3D	<code>Ext.CLIJ2_pushResultsTableColumn(image_name, column_name)</code>
	Pull to Results table		2D 3D	<code>Ext.CLIJ2_pullToResultsTable(image_name);</code>
	Push Array	[1,4,0,0,2], 3, 2, 1	2D 3D	<code>CLIJ2_pushArray(image_name, array, width, height, depth);</code>

Detailed documentation

CLIJ documentation can be found

- in CLIJ's dialogs under the menu Plugins > ImageJ on GPU (CLIJ2)
- Embedded in Fijis script editor – just start typinh
- and online: <https://clij.github.io/clij2-docs>

Installation instructions

- Install CLIJ2 by activating the "clij" and "clij2" update sites in Fiji.
- Commands listed as "CLIJx" are experimental should be handled with care. They may change or disappear at any point. To build reliable, reproducible workflows use CLIJ or CLIJ2 commands only.

