



Logical Image Processing System (LIPS) Version 3.0 Specifications

General		
1	Name of the software	Logical Image Processing System
2	Version	3.0
3	Revision	1
4	Media Format	CD-ROM
5	Product Number	PRCCVLLIPS3P0XXX
6	On-line help	HTML compiled
7	Documentation	License terms, Installation manual, User's guide, Specifications guide, Application Notes and Warranty.
8	Validation	Beta tested
9	License	Complete Package
10	License Quantity	01/05/10/20 users for each license
11	Company Web Site	www.pentagramresearch.com
12	Contact details	Telephone: +91-40-23533108, E-mail: commercial@pentagramresearch.com
13	On-line technical support	1000 – 1330 Hours (week days) 1400 – 1730 Hours (week days)
Technical		
1	System Type	IBM – PC compatibles
2	Software Type	Application (Image Processing)
3	Operating System	Microsoft Windows NT / 2000, Windows XP
4	Programming language used	Visual Studio.Net
5	Security measures	Hardware lock (USB)
6	Image formats	BMP/JPG
7	Display	Choice of built-in 12.1 Inch or 14.1 Inch (diagonal) TFT XGA active-matrix display, Support for millions of colors at 1024 x 768 pixel resolution, Support for resolution scaling to 800 x 600 pixel and 640 x 480 pixel resolution with millions of colors
8	Graphics Support	Any suitable graphics accelerator
9	Algorithms development	Developed using Cellular Logic Array Processing techniques and classical mathematical tools
10	Memory required	128MB
11	Storage space required	10MB (Minimum)
12	Processor Class	Pentium and its generics (500 MHz Minimum)
13	Peripheral support	Printer compatible (Laser printer and Ink Jet printer)
14	Other Requirements	CDD, Graphics card, SVGA Color monitor

Functional		
1	File open	Bitmap and JPG images could be opened by this facility
2	File save	Images in the current window could be saved as such or as a different file
3	File print	Image in the current window could be printed by a laser printer or by an ink jet printer
4	Capture	This is a print screen facility
5	Undo	Undoing could be carried out on processed image sequentially
6	Redo	After undoing one can redo (get back) the processed images sequentially.
7	Copy	Image in the current window could be saved temporarily in the clip board for future use
8	Paste	Image saved in the clip board could be pasted on a new window
9	Histogram	Frequency of occurrence of R, G and B color components values from 0 to 255 are depicted in separate slots.
10	Zoom-in	Image size is apparently enlarged by window stretching.
11	Zoom-out	Image size is apparently diminished by window shrinking.
12	Lens	Selected portion of the image is magnified by a lens of circular or rectangular shape, of varied diameters and of varied magnification.
13	Add	Point wise addition of a number to all pixel values is carried out.
14	Subtract	Point wise subtraction of a number from all pixel values is carried out.
15	Multiply	Point wise multiplication of a number from all pixel values is carried out.
16	Logarithmic	Point wise logarithm of all pixel values is carried out.
17	Exponential	Point wise exponentiation of all pixel values is carried out.
18	Square	Point wise squaring of all pixel values is carried out.
19	Absolute	Point wise difference of a pixel value and half of the maximum value in the image is carried out for all pixels.
20	Gamma	Each and every pixel in the image is gamma corrected.
21	Highlight	All those image pixel values in a range k_1 to k_2 are replaced by a single value k_3 and the image pixel values which are out of this range are kept in tact. By this, one can segment regions whose pixel values are in a required color range or gray range.
22	Threshold	All those image pixel values in the range k_1 to k_2 are replaced by the maximum value, that is, the white (255, 255, 255), and the image pixel values which are out of this range are made black (0, 0, 0). By this, one can segregate regions whose pixel values are in a required color range or gray range.
23	Equalize	Histogram equalization is carried out on a given image.
24	Dyadic add	Two images are added point wise
25	Dyadic subtract	Two images are subtracted point wise
26	Dyadic multiply	Two images are multiplied point wise
27	Dyadic divide	Two images are divided point wise
28	Dyadic maximum	Point wise maximum of two images are given in the resulting image.
29	Dyadic minimum	Point wise minimum of two images are given in the resulting image.
30	ContourRajan1	Contouring of an image using Rajan1 operator
31	ContourRajan2	Contouring of an image using Rajan2 operator

Functional specifications continued...		
32	ContourRajan3	Contouring of an image using Rajan3 operator
33	ContourRobert	Contouring of an image using Robert operator
34	ContourSobel1	Contouring of an image using Sobel1 operator
35	ContourSobel2	Contouring of an image using Sobel2 operator
36	ContourSobel3	Contouring of an image using Sobel3 operator
37	ContourPrewitt1	Contouring of an image using Prewitt1 operator
38	ContourPrewitt2	Contouring of an image using Prewitt2 operator
39	Dilate	Morphological operation of dilation could be carried out in an image with different structuring elements
40	Erode	Morphological operation of erosion could be carried out in an image with different structuring elements
41	Open	Morphological operation of opening could be carried out in an image with different structuring elements
42	Close	Morphological operation of closing could be carried out in an image with different structuring elements
43	ContourByDilation	Contouring of an image by subtracting original image from its one-pixel dilated version.
44	ContourByErosion	Contouring of an image by subtracting its one-pixel eroded version from the original image.
45	Filter	A sequence of dilation and erosion operations (filtering) could be carried out on an image.
46	SegmentByCellular	Segmentation of an image could be carried out based on quantization of neighborhood pixel values.
47	SegmentByMean	Segmentation of an image could be carried out based on mean of neighborhood pixel values.
48	SegmentByMedian	Segmentation of an image could be carried out based on median of neighborhood pixel values.
49	HighPassMask1	2-D High pass filtering of an image using mask 1
50	HighPassMask2	2-D High pass filtering of an image using mask 2
51	HighPassMask3	2-D High pass filtering of an image using mask 3
52	HighPassMask4	2-D High pass filtering of an image using mask 4
53	LowPassMask1	2-D Low pass filtering of an image using mask 1
54	LowPassMask2	2-D Low pass filtering of an image using mask 2
55	LowPassMask3	2-D Low pass filtering of an image using mask 3
56	LowPassMask4	2-D Low pass filtering of an image using mask 4
57	LowPassMask5	2-D Low pass filtering of an image using mask 5
58	KirschFilterMask1	2-D filtering of an image using Kirsch filter mask 1
59	KirschFilterMask2	2-D filtering of an image using Kirsch filter mask 2
60	KirschFilterMask3	2-D filtering of an image using Kirsch filter mask 3
61	KirschFilterMask4	2-D filtering of an image using Kirsch filter mask 4
62	KirschFilterMask5	2-D filtering of an image using Kirsch filter mask 5
63	KirschFilterMask6	2-D filtering of an image using Kirsch filter mask 6
64	KirschFilterMask7	2-D filtering of an image using Kirsch filter mask 7
65	KirschFilterMask8	2-D filtering of an image using Kirsch filter mask 8
66	PrewittFilterMask1	2-D filtering of an image using Prewitt filter mask 1
67	PrewittFilterMask2	2-D filtering of an image using Prewitt filter mask 2
68	PrewittFilterMask3	2-D filtering of an image using Prewitt filter mask 3

Functional specifications continued...		
69	PrewittFilterMask4	2-D filtering of an image using Prewitt filter mask 4
70	PrewittFilterMask5	2-D filtering of an image using Prewitt filter mask 5
71	PrewittFilterMask6	2-D filtering of an image using Prewitt filter mask 6
72	PrewittFilterMask7	2-D filtering of an image using Prewitt filter mask 7
73	PrewittFilterMask8	2-D filtering of an image using Prewitt filter mask 8
74	PrewittFilterMask9	2-D filtering of an image using Prewitt filter mask 9
75	SobelFilterMask1	2-D filtering of an image using Sobel filter mask 1
76	SobelFilterMask2	2-D filtering of an image using Sobel filter mask 2
77	SobelFilterMask3	2-D filtering of an image using Sobel filter mask 3
78	SobelFilterMask4	2-D filtering of an image using Sobel filter mask 4
79	SobelFilterMask5	2-D filtering of an image using Sobel filter mask 5
80	SobelFilterMask6	2-D filtering of an image using Sobel filter mask 6
81	SobelFilterMask7	2-D filtering of an image using Sobel filter mask 7
82	SobelFilterMask8	2-D filtering of an image using Sobel filter mask 8
83	FalerFilterMask1	2-D filtering of an image using Faler filter mask 1
84	FalerFilterMask2	2-D filtering of an image using Faler filter mask 2
85	FalerFilterMask3	2-D filtering of an image using Faler filter mask 3
86	FalerFilterMask4	2-D filtering of an image using Faler filter mask 4
87	LaplacianFilterMask1	2-D filtering of image using Laplacian filter mask 1
88	LaplacianFilterMask2	2-D filtering of image using Laplacian filter mask 2
89	LaplacianFilterMask3	2-D filtering of image using Laplacian filter mask 3
90	LaplacianFilterMask4	2-D filtering of image using Laplacian filter mask 4
91	LaplacianFilterMask5	2-D filtering of image using Laplacian filter mask 5
92	EdgeEnhancementEast	East wise edge enhancement of an image
93	EdgeEnhancementWest	West wise edge enhancement of an image
94	EdgeEnhancementNorth	North wise edge enhancement of an image
95	EdgeEnhancementSouth	South wise edge enhancement of an image
96	EdgeEnhancementNorthEast	North-East wise edge enhancement of an image
97	EdgeEnhancementSouthEast	South-East wise edge enhancement of an image
98	EdgeEnhancementSouthWest	South-West wise edge enhancement of an image
99	EdgeEnhancementNorthWest	North-West wise edge enhancement of an image
100	LineEnhancementEastWest	East-West wise line enhancement of an image
101	LineEnhancementNorthSouth	North-South wise line enhancement of an image
102	LineEnhancementNorthEast SouthWest	North East – South West wise line enhancement of an image

Functional specifications continued...		
103	LineEnhancementNorthWest SouthEast	North West – South East wise line enhancement of an image
104	EmbossHorizontal	Given image could be embossed in horizontal direction
105	EmbossVertical	Given image could be embossed in vertical direction
106	EmbossDiagonal	Given image could be embossed in diagonal directions
107	EmbossEast	Given image could be embossed in eastern direction
108	EmbossNorth	Given image could be embossed in northern direction
109	EmbossNorthWest	Given image could be embossed in north-western direction
110	EngraveHorizontal	Given image could be engraved in horizontal direction
111	EngraveVertical	Given image could be engraved in vertical direction
112	EngraveDiagonal	Given image could be engraved in diagonal directions
113	EngraveWest	Given image could be engraved in western direction
114	EngraveSouth	Given image could be engraved in southern direction
115	CreateMask	An empty mask (3X3 empty window) would be created. Entries of window coefficients could be made. Image could be processed using such custom masks.
116	ColorFilterAdd	Any RGB color value could be added to all pixels of a given image.
117	ColorFilterSubtract	Any RGB color value could be subtracted from all pixels of a given image.
118	ExtractDefault	Range of pixel values between a specified minimum value and a maximum value in an image would be assigned the value 0 and the remaining pixels the value 255.
119	ExtractSolid	Range of pixel values between a specified minimum value and a maximum value in an image would be assigned the value 0 and the remaining pixels would be retained as such.
120	Mirror	Lateral inversion (mirror) would be carried out in an image.
121	Flip	Vertical inversion (flip) would be carried out in an image.
122	RotateLeft	Image would be rotated to left by 90 degrees.
123	RotateRight	Image would be rotated to right by 90 degrees.
124	Rotate	Image would be rotated to right by any desired angle.
125	EllipticTransform	Elliptic transform of an image would be carried out by mapping the rectangular coordinates of the given image to an ellipse of a desired major axis and a minor axis.
126	CircularTransform	Circular transform of an image would be carried out by mapping the rectangular coordinates of the given image to a circle
127	SubsampleHexagonal	Given image would be subsampled in a hexagonal pixel grid.
128	SubsampleRectangular	Given image would be subsampled in a rectangular pixel grid.
129	SubsampleAdd	Point wise addition of a number to all pixel values of a subsampled image.

Functional specifications continued...		
130	SubsampleSubtract	Point wise subtraction of a number from all pixel values of a subsampled image.
131	SubsampleMultiply	Point wise multiplication of a number from all pixel values of a subsampled image.
132	SubsampleLogarithmic	Point wise logarithm of all pixel values of a subsampled image.
133	SubsampleExponential	Point wise exponentiation of all pixel values of a subsampled image.
134	SubsampleSquare	Point wise squaring of all pixel values of a subsampled image.
135	SubsampleAbsolute	Point wise difference of a pixel value and half of the maximum value in the image is carried out for all pixels of a subsampled image.
136	SubsampleGamma	Each and every pixel of a subsampled image would be gamma corrected.
137	SubsampleHighlight	All those image pixel values in a range k1 to k2 are replaced by a single value k3 and the subsampled image pixel values which are out of this range are kept in tact. By this, one can segment regions whose pixel values are in a required color range or gray range.
138	SubsampleThreshold	All those image pixel values in the range k1 to k2 are replaced by the maximum value, that is, the white (255, 255, 255), and the subsampled image pixel values which are out of this range are made black (0, 0, 0). By this, one can segregate regions whose pixel values are in a required color range or gray range.
139	SubsampleEqualize	Histogram equalization is carried out on a given subsampled image.
140	SubsampleDyadicAdd	Two subsampled images are added point wise
141	SubsampleDyadicSubtract	Two subsampled images are subtracted point wise
142	SubsampleDyadicMultiply	Two subsampled images are multiplied point wise
143	DyadicDivide	Two subsampled images are divided point wise
144	DyadicMaximum	Point wise maximum of two subsampled images are given in the resulting image.
145	DyadicMinimum	Point wise minimum of two subsampled images are given in the resulting image.
146	SubsampleContourRajan1	Contouring of a subsampled image using Rajan1 operator
147	SubsampleContourRajan2	Contouring of a subsampled image using Rajan2 operator
148	SubsampleExtractDefault	Range of pixel values between a specified minimum value and a maximum value in a subsampled image would be assigned the value 0 and the remaining pixels the value 255.
149	SubsampleExtractSolid	Range of pixel values between a specified minimum value and a maximum value in a subsampled image would be assigned the value 0 and the remaining pixels would be retained as such.
150	Corners	Points of intersection of lines would be extracted from a given image.

Functional specifications continued...

151	Curves	Nonlinear curve segments would be extracted from a given image.
152	Dots	Isolated points would be extracted from a given image.
153	Lines	Straight lines would be extracted from a given image.
154	TexturesRank1	Texture patterns in North-West to South-East direction are enhanced in a given image.
155	TexturesRank2	Texture patterns in North to South direction are enhanced in a given image.
156	TexturesRank3	Texture patterns in North-East to South-West direction are enhanced in a given image.
157	TexturesRank4	Texture patterns in East to West direction are enhanced in a given image.
158	VolumeFraction	Volume fraction of a desired region in a given image would be evaluated.
159	Lighten	The intensity of a given image would be increased by adding a default value to each and every pixel in the image.
160	Darken	The intensity of a given image would be decreased by adding a default value to each and every pixel in the image.
161	Contrast	Low contrast images are due to poor lighting conditions of the imaging sensor. Such images would be processed by a typical contrast stretching algorithm.
162	Soften	Image softening would be carried out by finding the average of all nine pixels and by replacing the central pixel with the average value.
163	Sharpen	Image sharpening would be carried out by subtracting the softened image from the original image.
164	PseudoColors	Pseudo coloring is carried out on gray images by segmenting the given image and painting each region with a color.
165	Negative	Negative of a given color image is obtained by subtracting the pixels from the maximum values of R, G and B present in the image.
166	Solarize	Dark regions below a specified pixel value in an image would be transformed into negative while the remaining pixel values kept in tact.
167	SplitToRGB	A color bitmap image would be split to its components of R, G and B color image files with gray shade.
168	SplitToHSL	A color bitmap image would be split to its components of H, S and L image files with gray shade.
169	SplitToYUV	A color bitmap image would be split to its components of Y, U and V image files with gray shade.
170	SplitToYIQ	A color bitmap image would be split to its components of Y, I and Q image files with gray shade.
171	SplitToXYZ	A color bitmap image would be split to its components of X, Y and Z image files with gray shade.
172	SplitToCMYK	A color bitmap image would be split to its components of C, M, Y and K color image files with gray shade.
173	GrayScale	A color image would be converted to a gray image. Note that the saved gray image cannot be converted back to its original color image.

Functional specifications continued...

174	FloydSteinberg Dither	A color image would be converted to a dithered binary image using Floyd-Steinberg algorithm.
175	OrderedDither	A color image would be converted to a dithered binary image using Ordered Dithering algorithm.
176	Text	Any text message with desired font and color could be written on a given image.
177	StraightLine	A straight line can be drawn on an image using this tool.
178	Rectangle	Any desired rectangle can be drawn on an image using this tool.
179	Polygon	Any desired polygon can be drawn on an image using this tool.
180	LineLength	This tool could be used to draw a straight line in any desired direction on an image whose length would be shown in a dialog box.
181	Pencil	This tool could be used to scratch any arbitrary line (curve) in any desired direction on an image whose length would be shown in a dialog box.
182	Terrain	This tool opens a small window in which the R, G and B values of all pixels of the image covered by a line drawn on it would be shown at various heights depending on their values. The reference point would be the first pixel on the line drawn on the image.
183	NewWindow	The given image would be copied on a new window.
184	Cascade	The window of the given image would be stretched partially so that any number of images of varied dimensions could be cascaded with the original image.
185	Tile	The window of the given image would be stretched fully so that any number of images of varied dimensions could be tiled with the original image.
186	CloseAll	This command closes all images shown on the monitor.
187	Duplicate	This command duplicates a given image in a different window.
188	HelpFile	This command invokes the on-line help manual of the software package LIPS version 3.0
189	AboutLIPS	This command pops up a dialog box, which provides some information related to the development of the software package.