Edinburgh
Mouse
Atlas
Project

Medical
Research
Council
MRC
Human Genetics Unit

EMAP Technical Report

MAPaint User Manual and Help Pages

Author:	**Richard Baldock**
Email:	`R.Baldock@hgu.mrc.ac.uk`
Date:	**January 2001**
Distribution:	**Open distribution is permitted**
Status:	**Draft**
Source:	**Report/000013-MAPaint_help**
Project:	**DG401**
Report number:	`EMAP/Report/000013`
Keywords:	

EMAP is a research programme at the MRC Human Genetics Unit in collaboration with Biomedical Sciences, University of Edinburgh. Contact Dr Duncan Davidson or Dr Richard Baldock at MRC Human Genetics Unit, Crewe Road, Edinburgh, EH4 2XU, UK.

Abstract

This is a user manual and help pages combined for the program MAPaint which is for manual and image-processing assisted segmentation of a 3D voxel image (reference image) into arbitrary, 3d, spatially non-overlapping regions (domains). There is a more detailed document of the underlying program and geometry available in html or postscript.

The program has been developed as part of the Edinburgh Mouse Atlas and Gene Expression Database Project at the MRC Human Genetics unit in Edinburgh (see http://genex.hgu.mrc.ac.uk/)

Contents

1 Introduction

The HGU paint program has been developed for the special purpose of ”painting” or segmenting an arbitrary 3-dimensional domain within a given 3-dimensional voxel image. The basic idea is that the user can view a slice of the underlying voxel data at any arbitrary position and angle. In fact there can be as many such views displayed simultaneously as are required (see the Views Menu). Each view displays a number of voxels from the underlying reference image and the user can mark or paint the voxels as belonging to the domain of interest. As soon as the voxels have been painted in one of the views then they will show as painted in all other views if visible. By updating the view and continuing the process it is possible to define an arbitrary 3D domain within the reference image.

It is possible to delineate more than one domain at a time and each will appear in its own colour, with a maximum of thirty-two possible colours (see Domain Menu). In general the different domains will have common boundaries and for convenience boolean rules are imposed on the domains determined by a dominance ordering, such that domains of higher dominance will always “win” over domains of lower dominance. The dominance ordering can be modified interactively using a drag and drop interface (see Domain Controls). It is also possible to set a mask from which all domains will be excluded (see Domain Controls). Individual colours and the grey-level colour map can be redefined (see Colomap Controls).

Defining arbitrary regions “by hand” is very time consuming and required intense concentration and so a number of “power-assist” tools have been provided (see Painting Tools). These use image-processing techniques to assist the manual segmentation process.

A reference document is available which provides a fuller description of the program and give the details of the view-geometry and image processing and analysis functions, in html or postscript.

2 Basic Concepts and Terms

The underlying assumption within paint is that the user wishes to define an arbitrary region or regions of 3D space within the space defined by a 3D voxel image termed the reference image. These regions are termed domains and it is assumed that during any one session the domains to be defined do not overlap although they may have a common boundary. It also assumed that the spatial resolution of the reference image defines the resolution of the domains i.e. the domain boundaries are defined by the voxel boundaries of the reference image.

In paint version 1.0 the reference image must be a grey-level image and the native image format is woolz. This means that the image is not restricted to a rectangular array of voxel values but to a set of values defined within an arbitrary region of 3D space. The only restriction on the domains is that they can only be defined within this 3D region i.e. the user is not allowed to ”paint” on regions not in the reference image.

As each domain is painted this modifies an internal record of the painted voxels but does not modify the reference image. Domains and the reference image are independent 3D image structures defined which map into the same co-ordinate frame but which can be independently read, written and modified. To save a defined domain it is necessary to select it and then write to disc. Attempting to quit paint with modified but unsaved domains will result in a warning message which if ignored will result in a loss of the domain when the program quits. The saved domains are woolz binary domain objects which can be externally modified by the various woolz filters and programs.

Within paint the domains are ordered in a dominance hierarchy. The effect of dominance is that domains of higher dominance will always "win" over domains of lower dominance with the exception of when a domain is read in using the read-domain menu option (see Domain Menu). In this case the domain read in is temporarily placed at the top of the dominance hierarchy.

To define or modify a domain the user selects a view-window and then uses the painting tools to define the domain voxels. The selected view-window becomes highlighted and only one window can be used for painting. If painting in a different view is required then the first view must be de-selected by confirming that painting has finished in that window (right-button press). As soon as painting in a window or plane finishes, all other view windows will be updated if they have overlapping voxels.

Typical usage of the program is to define a domain in one plane then to move to the next parallel plane and define the voxels for the same domain. In many cases these will have almost the same 2D distribution therefore paint has the facility to propagate the 2D domain defined in one planar section to the next. Propagation is user controllable and will provide warning (if requested) if existing domains will be modified by the propagation process. This process is supported by the auto-increment option which, if on, will automatically move the view to the next plane when painting finishes (see Domain Menu).

Each domain appears as a coloured region overlaid on the grey-values of the reference image. The first five colours appear as "wash" overlays through which the grey-values can be seen. The remaining colours (27) are solid but when deleted or temporarily "lifted" will reveal the underlying reference image. The number of colours is by default 10 but this can be modified by setting an X11 resource. Practice has shown that the wash colours are the most useful because keeping the reference image visible makes editing easier but there is a limitation to 5 because the paint program has been designed to work on an 8-bit display. The colour of each overlay, the intensity of the wash and the grey-level look-up table of the reference image can all be adjusted interactively (see Colormap Dialog).

Paint allows the domain voxels to be defined within any arbitrary section view of the reference image. This view is defined by a set of viewing-coordinates which comprise a fixed point (3D), a viewing direction and a distance. The section through the reference image is perpendicular to the viewing direction and at the given distance from the fixed point. These co-ordinates and transformations are defined in the reference document. The viewing direction is defined by two angles phi and theta which can be thought of as pitch and yaw respectively. If these angles are kept constant then adjusting distance will display parallel planes from the reference image. Initially the fixed point is set at 3D co-ordinate (0,0,0) but this can be reset at any time. Rotation is always about the fixed point therefore if the user wishes to navigate by finding a point known to be in the required section, this point can be set as the fixed point and then adjusting the viewing direction will result in planes passing through that fixed point. A further navigation tool id provided by a second fixed point in which case the degrees of freedom are reduced to a single angle, psi, which defines rotation around the fixed line.

Given the viewing co-ordinates there is still ambiguity about orientation of the displayed section on the screen. Two options have been defined (with only one currently available) which are "walking round the statue" and "up is up", These are also detailed in the reference document.

3 Installing MAPaint

MAPaint has been developed in C with the X11/Motif window system. It will therefore only work with Unix systems. There are earlier versions of the program which run on SGI but this version has only been tested on Solaris 2.5 or higher. If versions are required for other architectures then please contact the author. Source can be made available for compilation on other Unix systems.

3.1 System Requirements

Operating system: Solaris 2.5 or higher.

CPU: in principle any that can run Solaris 2.5+, in practice a Sparc 10 or better is required for a reasonably interactive response. The current development machine is a Sun Ultra 10.

RAM: for interactive sectioning the available RAM should be about double to volume of the 3D voxel image. For the 9-day mouse embryo (approx 10MBytes) on a Sun workstation, 64MBytes is sufficient (the window system occupies about half!).

Disk space: The binary occupies about 2.2MBytes. In normal operation the autosave option requires space equal to the size of the input grey-level voxel image. The autosave feature can be switched off.

Software environment: MAPaint requires an X11R5 or higher X-Window server either with OpenGL extensions or with the Mesa OpenGL dynamic libraries. MAPaint will also use the Sun XIL extensions if presents and installed.

3.2 Obtaining the Program

Contact the author or from the WWW site http://genex.hgu.mrc.ac.uk/ where it is available for download as a compressed tar file.

3.3 Installation

The tar file should be uncompressed and unpacked in some suitable place, e.g. /opt/MouseAtlas/MAPaint. The binary can then be copied as required, e.g. to /opt/MouseAtlas/bin with users' executable path appropriately modified. If the Mesa software implementation of OpenGL is required then this can be obtained from the Mouse Atlas site or directly from http://ssec.wisc.edu/ brianp/Mesa.html.

4 Running MAPaint

MAPaint is simply run from the command line, usually in the directory containing the reference voxel image or where the painted domains are to be stored. MAPaint can also be set up to run from window manager menu but the user will then be required to switch to the desired directory. MAPaint does not use any user preferences files (except X11 resources) therefore has no mechanism for restoring a previous session automatically. In due course this may be possible via the Save/Rstore facility but this must be set and recovered "by hand".

4.1 Command Line Arguments

Apart from X11 resources, including all the application specific resources below which can be set with the -xrm flag, there are only two additional command line arguments.

Synopsis:

MAPaint [X11 arguments] [<reference image>[<feedback image>]]

Options:

<reference image>	woolz format grey-level 3D reference image
<feedback image>	woolz format image to generate the 3D feedback

4.2 X11 Resources

There are a few resources to modify the behavior of paint, a few more that modify the layout and a large number that simply define the menu labels etc. These are all listed below with the resources likely to be of most interest in the first section. This list represents the fallback resources list encoded in the program. Note if these resources are to be used in the users' .XDefaults or .XResources file each listed resource name must be prepended with "MAPaint".

4.2.1 Behavior and Layout Resources

Resource	Default	Comment
.geometry	258x332	
.autosaveTime	900	Autosave can be switched off only when the program is running.
.autosaveDir	/tmp	Save directory for the autosave file. Set to /tmp for efficiency on Solaris machines (/tmp is memory mapped) but to ensure the file is saved if the machine has to rebooted set this to a directory not automatically cleared.
.autosaveFile	MAPaintAutosave	This file name will have the process id appended to generate a unique filename. Note by default the autosave file is removed when the program quits.
*background	grey85	
*XmScrollBar.background	grey95	
*fontList	-*-helvetica-bold-r-*-*-12-*-*-*-*-*-*-*	
*XmText.fontList	8x13	
*XmTextField.fontList	8x13	
*XmList.fontList	8x13	
*XmText.background	grey95	
*XmTextField.background	grey95	
*XmList.background	grey95	
*menubar.XmCascadeButton.marginWidth	1	
*XmToggleButtonGadget.selectColor	red	
*XmToggleButtonGadget.visibleWhenOff	True	
*XmToggleButtonGadget.indicatorSize	14	
*XmToggleButton.selectColor	red	
*XmToggleButton.visibleWhenOff	True	
*XmToggleButton.indicatorSize	14	
*XmFrameWidget.shadowThickness	3	
*XmForm.horizontalSpacing	4	
*XmForm.verticalSpacing	4	
*XmFrame.borderWidth	0	
*gamma	1.0	Colormap gamma value
*invert	0	Select inverted colormap
*numSolidOverlays	5	Number of "solid" overlay colours (max 32).
*numWashOverlays	5	Number of "wash" overlay colours (max 5).

4.2.2 3D View, 3D Section and OpenGL options

Resource	Default	Comment
*work_area.width	300	Width of the 3D feedback view
*work_area.height	300	Heaight of the 3D feedback view
*work_area*doublebuffer	True	OpenGL behavior
*work_area*allocateBackground	True	
*work_area*background	black	
*work_area*allocateotherColors	True	
*work_area*installColormap	True	
*HGU_DrawingArea.visual	PseudoColor	
*HGU_DrawingArea.depth	8	
*HGU_DrawingArea.background	white	

4.2.3 File Menu

Resource	Default	Comment
*file_menu.labelString	File	
*file_menu.mnemonic	F	
*file_menu.borderWidth	0	
*file_menu*new_obj.labelString	New	
*file_menu*new_obj.mnemonic	N	
*file_menu*read_obj.labelString	Open reference	
*file_menu*read_obj.mnemonic	O	
*file_menu*write_obj.labelString	Write reference	
*file_menu*write_obj.mnemonic	W	
*file_menu*obj_props.labelString	Ref. properties	
*file_menu*obj_props.mnemonic	P	
*file_menu*theiler_stage.labelString	Theiler stage	
*file_menu*theiler_stage.mnemonic	T	
*file_menu*theiler_setup.labelString	Theiler setup	
*file_menu*save_state.labelString	Save state	
*file_menu*save_state.mnemonic	S	
*file_menu*restore_state.labelString	Restore	
*file_menu*restore_state.mnemonic	R	

4.2.4 Anatomy Menu

Resource	Default	Comment
*anatomy_menu.labelString	Anatomy	
*anatomy_menu.mnemonic	A	

4.2.5 Domain Menu

Resource	Default	Comment
*domain_menu.labelString	Domain	
*domain_menu.mnemonic	D	
*domain_menu.borderWidth	0	
*domain_menu*select.labelString	Select	
*domain_menu*select.mnemonic	S	
*domain_menu*select.borderWidth	0	
*domain_menu*controls.labelString	Controls...	
*domain_menu*controls.mnemonic	C	
*domain_menu*auto_increment.labelString	Auto-increment	
*domain_menu*auto_increment.mnemonic	A	
*domain_menu*auto_increment.borderwidth	0	
*domain_menu*propogate.labelString	Propogate	
*domain_menu*propogate.mnemonic	P	
*domain_menu*propogate.borderwidth	0	
*domain_menu*read_domain.labelString	Read domain	
*domain_menu*read_domain.mnemonic	R	
*domain_menu*write_domain.labelString	Write domain	
*domain_menu*write_domain.mnemonic	W	
*domain_menu*save_domain.labelString	Save domain	
*domain_menu*save_domain.mnemonic	a	
*domain_menu*clear_domain.labelString	Clear domain	
*domain_menu*clear_domain.mnemonic	l	
*domain_menu*clear_all_domains.labelString	Clear all domains	
*domain_menu*threed_display_domain.labelString	Display 3D domain	
*domain_menu*threed_display_domain.mnemonic	3	
*domain_menu*write_paint_volume.labelString	Write painted volume	
*domain_menu*write_paint_volume.mnemonic	I	
*domain_menu*select*borderWidth	0	

4.2.6 View Menu

Resource	Default	Comment
*view_menu.labelString	View	
*view_menu.mnemonic	V	
*view_menu.borderWidth	0	
*view_menu*x_y_view.labelString	X-Y view...	
*view_menu*x_y_view	nemonic X	
*view_menu*y_z_view.labelString	Y-Z view...	
*view_menu*y_z_view	nemonic Y	
*view_menu*z_x_view.labelString	Z-X view...	
*view_menu*z_x_view	nemonic Z	
*view_menu*arbitrary_view.labelString	Arbitrary...	
*view_menu*arbitrary_view	nemonic A	

4.2.7 Options Menu

Resource	Default	Comment
*options_menu.labelString	Options	
*options_menu.mnemonic	O	
*options_menu.borderWidth	0	
*options_menu*paint_tools_2D.labelString	Paint tools (2D)	
*options_menu*paint_tools_3D.labelString	Paint tools (3D)	
*options_menu*tool_controls.labelString	Tool controls...	
*options_menu*tool_controls.mnemonic	T	
*options_menu*domain_review.labelString	Domain review...	
*options_menu*domain_review.mnemonic	D	
*options_menu*domain_surgery.labelString	Domain surgery...	
*options_menu*domain_surgery.mnemonic	S	
*options_menu*auto_segment.labelString	Auto segment...	
*options_menu*auto_segment..mnemonic	A	
*options_menu*colormap.labelString	Colormap...	
*options_menu*colormap.mnemonic	C	
*options_menu*autosave_opts.labelString	Autosave...	
*options_menu*autosave_opts.mnemonic	A	

4.2.8 Miscellaneous Menu

Resource	Default	Comment
*misc_menu.labelString	Misc	
*misc_menu.mnemonic	M	
*misc_menu*macro.labelString	Macro	
*misc_menu*macro*start_record.labelString	Start Record	
*misc_menu*macro*start_record.mnemonic	S	
*misc_menu*macro*end_record.labelString	End record	
*misc_menu*macro*end_record.mnemonic	E	
*misc_menu*macro*replay.labelString	Replay	
*misc_menu*macro*replay.mnemonic	R	
*misc_menu*macro*loop.labelString	Loop	
*misc_menu*macro*loop.mnemonic	L	
*misc_menu*macro*quit_loop.labelString	Quit loop	
*misc_menu*macro*quit_loop.mnemonic	Q	
*misc_menu*save_restore.labelString	Save/Restore	
*misc_menu*save_state.labelString	Save State	
*misc_menu*restore_state.labelString	Restore	
*misc_menu*save_restore*save_state.labelString	Save State	
*misc_menu*save_restore*restore_state.labelString	Restore	

4.2.9 Help Menu

Resource	Default	Comment
*helpDefaultUrl	paint.1.0/paint.html	
*helpIndexUrl	paint.1.0/paint.html#index	
*helpVersionUrl	paint.1.0/paint.html#version	
*help_menu.labelString	Help	
*help_menu.mnemonic	H	
*help_menu.borderWidth	0	
*help_menu*on_program.labelString	Paint	
*help_menu*on_context.mnemonic	P	
*help_menu*on_context.labelString	On context	
*help_menu*on_context.mnemonic	c	
*help_menu*on_windows.labelString	On windows	
*help_menu*on_windows.mnemonic	w	
*help_menu*on_keys.labelString	On keys	
*help_menu*on_keys.mnemonic	k	
*help_menu*on_keys.sensitive	False	
*help_menu*index.labelString	Index	
*help_menu*index.mnemonic	I	
*help_menu*on_help.labelString	On help	
*help_menu*on_help.mnemonic	h	
*help_menu*tutorial.labelString	Tutorial	
*help_menu*tutorial.mnemonic	T	
*help_menu*tutorial.sensitive	False	
*help_menu*version.labelString	Version	
*help_menu*version.mnemonic	V	

4.2.10 File Dialogs

Resource	Default	Comment
*read_obj_dialog_popup.mappedWhenManaged	False	
*read_obj_dialog.dialogTitle	Read 3D Object	
*read_obj_dialog.pattern	*.wlz	
*read_obj_dialog.file_type.labelString	Select image type:	
*read_obj_dialog.file_type.borderWidth	0	
*write_obj_dialog_popup.mappedWhenManaged	False	
*write_obj_dialog.dialogTitle	Write 3D ReferenceObject	
*write_obj_dialog.pattern	*.wlz	
*write_obj_dialog.file_type.labelString	Select image type:	
*write_obj_dialog.file_type.borderWidth	0	
*object_props_dialog_popup.mappedWhenManaged	False	
*object_props_dialog.dialogTitle	Object Properties	
*object_props_dialog*control.horizontalSpacing	4	
*object_props_dialog*control.verticalSpacing	4	
*object_props_dialog*frame1.title1.labelString	Bounding box	
*object_props_dialog*form1*planes.labelString	Planes (z):	
*object_props_dialog*form1*lines.labelString	Lines (y):	
*object_props_dialog*form1*kols.labelString	Columns (x):	
*object_props_dialog*form1*planes_vals.labelString	0,100)	
*object_props_dialog*form1*lines_vals.labelString	(0,100)	
*object_props_dialog*form1*kols_vals.labelString	(0,100)	
*object_props_dialog*frame2.title2.labelString	Voxel parameters	
*object_props_dialog*form2*background.labelString	Background:	
*object_props_dialog*form2*x_size.labelString	X size:	
*object_props_dialog*form2*y_size.labelString	Y size:	
*object_props_dialog*form2*z_size.labelString	Z size:	
*XmFileSelectionBox*file_type*woolz.labelString	Woolz	
*XmFileSelectionBox*file_type*ics.labelString	ICS	
*XmFileSelectionBox*file_type*den.labelString	Stanford Density	
*XmFileSelectionBox*file_type*vff.labelString	Sunvision VFF	
*XmFileSelectionBox*file_type*pic.labelString	BioRad Confocal	
*XmFileSelectionBox*file_type*vtk.labelString	Visualization Toolkit VTK	
*XmFileSelectionBox*file_type*pgm.labelString	pgm	
*save_state_dialog.pattern	*.rsc	
*restore_state_dialog.pattern	*.rsc	

4.2.11 Domain Dialogs

Resource	Default	Comment
*domain_controls_dialog_popup.mappedWhenManaged	False	
*domain_controls_dialog.dialogTitle	Domain controls	
*domain_controls_dialog*dominance.labelString	Domain Dominance:	
*domain_controls_dialog*control.horizontalSpacing	5	
*domain_controls_dialog*control.verticalSpacing	5	
*domain_controls_dialog*dominance_frame.borderWidth	0	
*domain_controls_dialog*dominance_form.borderWidth	0	
*domain_controls_dialog*domain_1.labelString	1	
*domain_controls_dialog*domain_2.labelString	2	
*domain_controls_dialog*domain_3.labelString	3	
*domain_controls_dialog*domain_4.labelString	4	
*domain_controls_dialog*domain_5.labelString	5	
*read_domain_dialog_popup.mappedWhenManaged	False	
*read_domain_dialog.dialogTitle	Read 3D Domain	
*read_domain_dialog.pattern	*.wlz*	
*write_domain_dialog_popup.mappedWhenManaged	False	
*write_domain_dialog.dialogTitle	Write 3D Domain	
*write_domain_dialog.pattern	*.wlz*	
*write_paint_volume_dialog_popup.mappedWhenManaged	False	
*write_paint_volume_dialog.dialogTitle	Write 3D Painted Volume	
*write_paint_volume_dialog.pattern	*.wlz*	
*write_paint_volume_dialog.file_type.labelString	Select image type:	
*write_paint_volume_dialog.file_type.borderWidth	0	

4.2.12 View Dialog

Resource	Default	Comment
*view_dialog*save_section.labelString	Save	
*view_dialog*dismiss.labelString	Dismiss	
*view_dialog*help.labelString	Help	
*view_dialog*XmForm.horizontalSpacing	4	
*view_dialog*XmForm.verticalSpacing	4	
*view_dialog*XmFrame.borderWidth	0	
*view_dialog*scrolled_window.borderWidth	3	
*view_dialog*section_title_form.verticalSpacing	0	
*view_dialog*section_title_form.borderWidth	0	
*view_dialog*section_frame_title.labelString	Section:	
*view_dialog*section_frame_FB_toggle.labelString	filled 3D view	
*view_dialog*section_frame_FB_toggle.highlightThickness	0	
*view_dialog*section_frame_direction_toggle.labelString	view direction	
*view_dialog*canvas.borderWidth	0	
*view_dialog*distance_slider.labelString	Distance	
*view_dialog*controls_frame_title.labelString	Controls	
*view_dialog*view_scale.labelString	Scale:	
*view_dialog*view_mode.labelString	View Mode:	
*view_dialog*theta_slider.labelString	Yaw	
*view_dialog*phi_slider.labelString	Pitch	
*view_dialog*psi_slider.labelString	Line Rot.	
*view_dialog*up_test.labelString	Up Vector	
*view_dialog*fixed_1.labelString	Fixed Point	
*view_dialog*fixed_2.labelString	Fixed Line	
*view_dialog*up_vector.labelString	Up Vector	
*view_dialog*save_settings.labelString	Save	
*view_dialog*graphical_f1.labelString	Point select	
*view_dialog*textual_f1.labelString	Type in coords	
*view_dialog*fiducial_f1.labelString	Select fiducial	
*view_dialog*display_f1.labelString	Display	
*view_dialog*graphical_f2.labelString	Point define	
*view_dialog*textual_f2.labelString	Type in coords	
*view_dialog*fiducial_f2.labelString	Select fiducials	
*view_dialog*display_f2.labelString	Display	
*view_dialog*graphical_uv.labelString	Point define	
*view_dialog*textual_uv.labelString	Type in coords	
*view_dialog*fiducial_uv.labelString	Select fiducials	
*view_dialog*display_uv.labelString	Display	

4.2.13 Tool Controls Dialog

Resource	Default	Comment
*tool_control_dialog.dialogTitle	Painting Tool Controls	
*tool_control_dialog*XmForm.borderWidth	0	
*tracking_controls_form*XmFrame.borderWidth	0	
*tracking_parameters.labelString	Tracking Parameters:	
*tracking_controls_form*spacing.labelString	Spacing:	
*tracking_controls_form*range.labelString	Range:	
*tracking_controls_form*size.labelString	Size:	
*snake_cost_parameters.labelString	Snake Cost Parameters:	
*image_cost_parameters.labelString	Image Cost Parameters:	
*tracking_controls_form*nu_dist.labelString	Distance:	
*tracking_controls_form*nu_alpha.labelString	Angle:	
*tracking_controls_form*nu_kappa.labelString	Curvature:	
*edge_tracking_controls_form*XmFrame.borderWidth	0	
*tracking_parameters.labelString	Tracking Parameters:	
*edge_tracking_controls_form*spacing.labelString	Spacing:	
*edge_tracking_controls_form*range.labelString	Range:	
*edge_tracking_controls_form*size.labelString	Width:	
*snake_cost_parameters.labelString	Snake Cost Parameters:	
*image_cost_parameters.labelString	Image Cost Parameters:	
*edge_tracking_controls_form*nu_dist.labelString	Distance:	
*edge_tracking_controls_form*nu_alpha.labelString	Angle:	
*edge_tracking_controls_form*nu_kappa.labelString	Curvature:	
*edge_tracking_controls_form*nu_direction.labelString	Direction:	
*tablet_ref_points.labelString	Tablet Reference Points:	
*tablet_controls_form*x1.labelString	X1:	
*tablet_controls_form*y1.labelString	Y1:	
*tablet_controls_form*x2.labelString	X2:	
*tablet_controls_form*y2.labelString	Y2:	
*paint_draw_params.labelString	Draw/Paint Parameters:	
*paint_draw_controls_form*cursor.labelString	Cursor Type:	
*paint_draw_controls_form*paint_shape.labelString	Paint Shape:	
*paint_draw_controls_form*paint_border.labelString	aint Border	
*paint_draw_controls_form*blob_size.labelString	Paint size:	
*geometry_params.labelString	2D Geometry Parameters:	
*geometry_controls_form*geom_type.labelString	Object type:	
*geometry_controls_form*object_size.labelString	Object size:	
*geometry_controls_form*fixed_circle.labelString	Fixed circle	
*geometry_controls_form*fixed_square.labelString	Fixed square	
*geometry_controls_form*variable_circle.labelString	Interactive circle	
*geometry_controls_form*variable_square.labelString	Interactive square	
*threshold_params.labelString	Threshold Parameters:	
*threshold_controls_form*connect.labelString	Connectivity:	
*threshold_controls_form*range.labelString	Initial range:	
*morph_parameters.labelString	Morphological Operator Params:	
*morph_controls_form*struct_elem.labelString	SE type:	
*morph_controls_form*size.labelString	SE size:	
*affine_options.labelString	Affine Transform Options:	

4.2.14 Domain Review Dialog

Resource	Default	Comment
*review_dialog_popup.mappedWhenManaged	False	
*review_dialog.dialogTitle	Domain Review Controls	
*review_dialog*src_domain_title.labelString	Source domain:	
*review_dialog*domainSource.labelString	Source:	
*review_dialog*domainSource*review_domain.labelString	Review domain	
*review_dialog*domainSource*mapaint_domain.labelString	MAPaint domain	
*review_dialog*domainSource*read_domain.labelString	Read domain	
*review_dialog*src_domain.labelString	MAPaint domain:	
*review_dialog*domain_review_read.labelString	Read	
*review_dialog*dest_domain_title.labelString	Destination domain:	
*review_dialog*Discard.labelString	Discard	
*review_dialog*Greys.labelString	Greys	
*review_dialog*review_planes_title.labelString	Review planes:	
*review_dialog*review_scan_control_title.labelString	Scan control:	
*review_dialog*review_scale.labelString	Review scale:	
*review_dialog*segment_domain.labelString	Segment domain	
*review_dialog*segment_domain.set	True	
*review_dialog*segment_planes.labelString	Segment planes	
*review_dialog*segment_planes.set	True	
*read_review_obj_dialog.dialogTitle	Read Review Object	
*read_review_obj_dialog.pattern	*.wlz	

4.2.15 Surgery Dialog

Resource	Default	Comment
*surgery_dialog_popup.mappedWhenManaged	False	
*surgery_dialog.dialogTitle	Domain Surgery Controls	
*surgery_dialog*src_domain_title.labelString	Source domain:	
*surgery_dialog*src_domain.labelString	MAPaint domain:	
*surgery_dialog*surgery_option.labelString	Surgery option:	
*surgery_dialog*dest_domain_title.labelString	Destination domain:	
*surgery_dialog*Discard.labelString	Discard	
*surgery_dialog*Greys.labelString	Greys	

4.2.16 Colormap Dialog

Resource	Default	Comment
*colormap_dialog_popup.mappedWhenManaged	False	
*colormap_dialog.dialogTitle	Colormap controls	
*colormap_dialog*form.borderWidth	0	
*colormap_dialog*overlay.borderWidth	0	
*colormap_dialog*overlay.labelString	Select overlay:	
*colormap_dialog*gamma.labelString	Gamma:	
*colormap_dialog*low_thresh.labelString	Low:	
*colormap_dialog*high_thresh.labelString	High:	
*colormap_dialog*red.labelString	Red:	
*colormap_dialog*green.labelString	Green:	
*colormap_dialog*blue.labelString	Blue:	
*colormap_dialog*contrast.labelString	Contrast:	

4.2.17 Autosave Dialog

Resource	Default	Comment
*autosave_dialog_popup.mappedWhenManaged	False	
*autosave_dialog.dialogTitle	Autosave controls	
*autosave_dialog*autosave_time.labelString	utosave time(sec):	
*autosave_dialog*autosave_file.labelString	utosave save file:	
*autosave_dialog*autosave_recover_file.labelString	utosave recover file:	
*autosave_domains_dialog.dialogTitle	Autosave recover domains	

4.3 Input Data Formats

User images can be read into MAPaint to define the reference image, to define painted domains and as a domain to be reviewed. The reference and painted domains must be 3D data. 2D images can be read in for review but must be mapped directly onto a section through the reference image. Permitted input formats are:

Name	File extension	Source
woolz	.wlz	Woolz is the native image format for MAPaint http://genex.hgu.mrc.ac.uk/Software/woolz/
ICS	.ics,.ids	Standard for Image Cytometry Data Files (Cytometry, 11(1990)561-569)
Stanford Density	.den	http://white.stanford.edu/
Sunvision	.vff	3D voxel format for Sunvision now incorporated as part of AVS Express.
BioRad Confocal	.pic	3D format for the BioRad confocal microscope.
VTK	.vtk	Native format for the Visualisation Toolkit http://www.cs.rpi.edu/~martink
SLC	.slc	

If other image formats are required then use e.g. netpdm tools or the program “convert” from ImageMagick.

4.4 Output Data Formats

MAPaint allows image data to be output from the file menu, domain menu and the view dialog. These can be in 3D or 2D format with an enumerated sequence of 2D image if required. Output formats are:

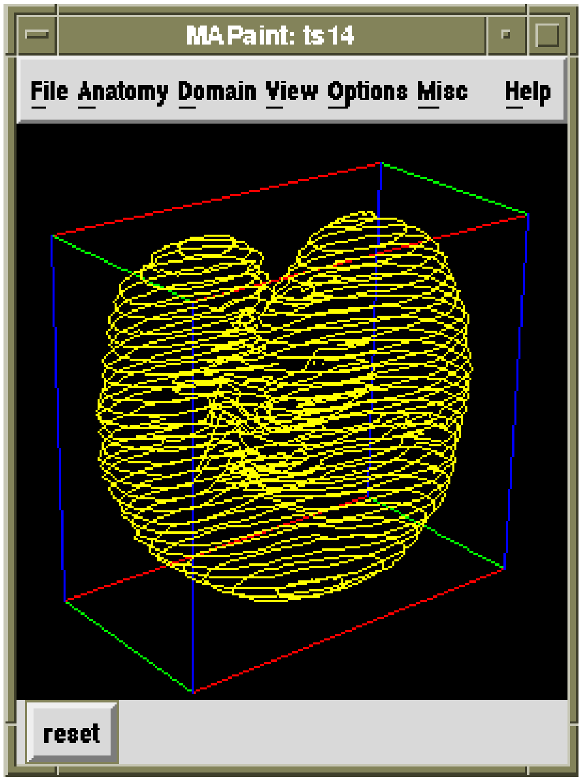
Name	File extension	Source
woolz	.wlz	Woolz is the native image format for MAPaint http://genex.hgu.mrc.ac.uk/Software/woolz/
ICS	.ics,.ids	Standard for Image Cytometry Data Files (Cytometry, 11(1990)561-569)
Stanford Density	.den	http://white.stanford.edu/
Sunvision	.vff	3D voxel format for Sunvision now incorporated as part of AVS Express.
BioRad Confocal	.pic	3D format for the BioRad confocal microscope.
VTK	.vtk	Native format for the Visualisation Toolkit http://www.cs.rpi.edu/~martink
SLC	.slc	
Netpbm	.pgm	Pixel grey image - netpbm standard. 3D image are written as an enumerated series of 2D images.
Bitmap	.bmp	Microsoft bitmap format. 3D image are written as an enumerated series of 2D images.

If other image formats are required then use e.g. netpdm tools or the program “convert” from ImageMagick.

5 The User Interface

MAPaint has been implemented in C with the X11 GUI implemented using the Motif widget set to provide a standard “look and feel”. Where the user interaction has had to be implemented directly in X11, the Motif style has been followed if possible.

5.1 Main Window

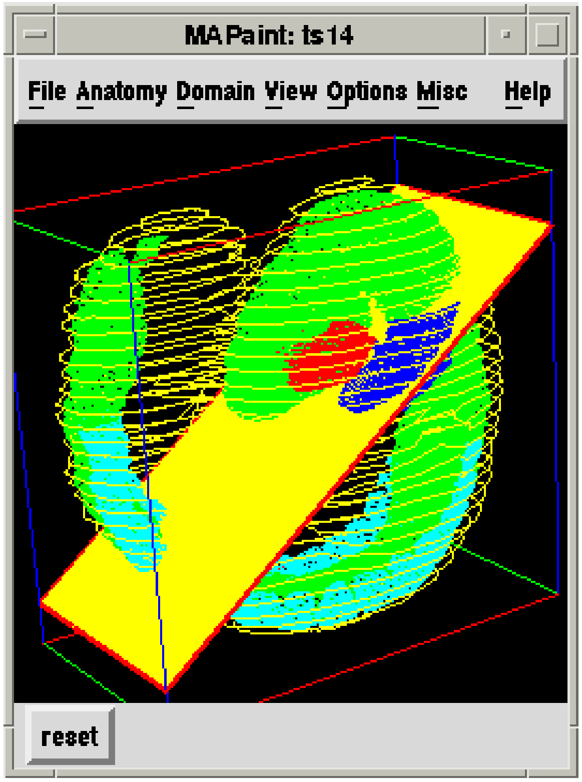


The paint program main window follows the standard Motif layout style of a menubar providing the pull-down menus, a work area, which displays a 3D representation of the current domains (if requested) and view planes, and a button bar for commonly used commands, in this version of MAPaint this is just a reset button for the 3D display.

The Anatomy menu (see section 5.2.2 becomes sensitive if the reference image has been read in from the Edinburgh Mouse Atlas CD-ROM using the Theiler stage option of the File menu. In this case a highly simplified 3D outline of the embryo is displayed (yellow lines in the figure);

The Help menu (see section 5.2.8 will access these help pages provided the help page URL resources are set correctly, Netscape browser is available and if the host computer is connected to the Internet.

5.1.1 3D Display



The 3D feedback window is fully interactive and allows the user to modify the view of the 3D scene by rotation, scaling and translation. The interaction model for rotation is that the scene is embedded in a sphere which the user can rotate by clicking the left mouse-button in the scene and dragging. If no views are selected for painting then the 3D view will move. If the mouse button is released while the mouse is still in motion then the 3D view will continue to move as if ”thrown”. Dragging the left-mouse button up/down with the shift key pressed results in a re-scaling of the scene and dragging the middle-mouse button will translate the scene. The reset button will return the viewing co-ordinates and scale to the default values.

When a view-window is opened the outline of the window will be displayed in the 3D scene and will move as the viewing parameters are modified. By this means the viewing window can be positioned relative to the displayed 3D domains. A view-window that has been selected for painting will be highlighted in red. This will also temporarily switch off the 3D interaction until painting has been completed.

The 3D view of domains is toggled on and off using the ”Display 3D domain” option on the Domain Menu.

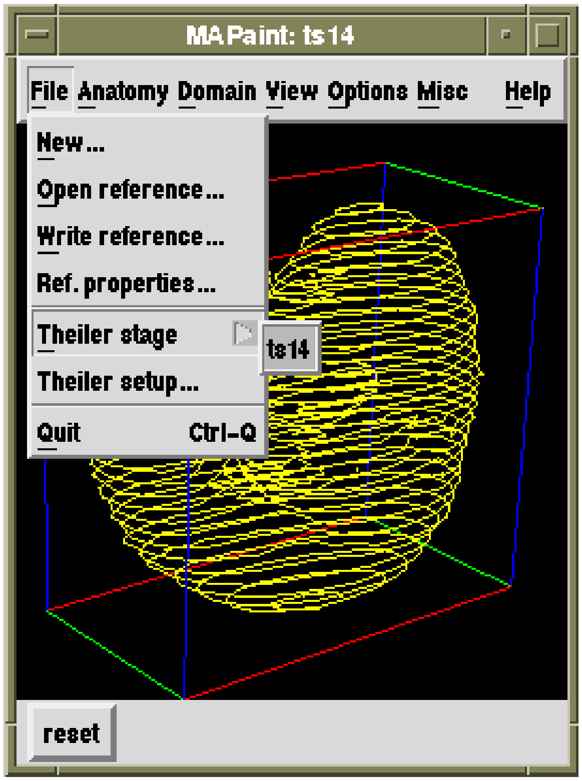
When the 3D view is set into motion the display will update after a certain period (?? ms). If the displayed domains are very complex this period may be shorter than the time required to recalculate the display. In this situation it is possible for the 3D feedback processing to hog all the available processing and interaction with the program becomes very slow. If this becomes a problem it is important to ensure that the 3D view is stationary. Only when a view is selected for painting are the 3D view updates paused.

5.2 Pull-down Menus

Some of the pull-down menus from the menu-bar can be “pinned”, i.e. made permanently visible on the screen. This is for convenience for commonly used actions, menus that can be pinned have a dotted line “tear-off” at the top of menu pop-up. The menu windows can be removed from display by double-clicking the top-left close button of the menu window.

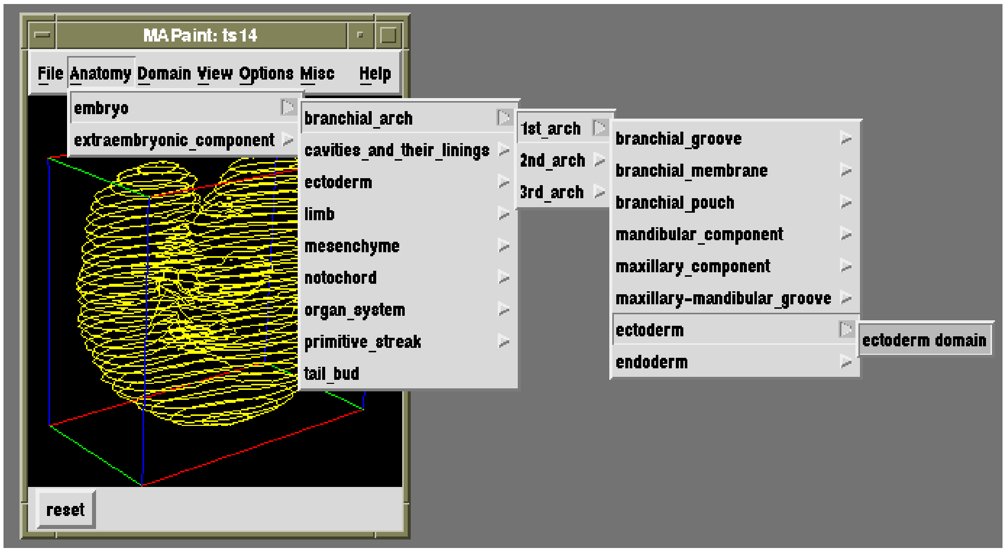
Within each menu, buttons that result in a further menu are indicated by an arrow to the right of the label, buttons that result in a dialog window are indicated by "..." following the text.

5.2.1 File Menu



The file menu provides for read and write of the reference image. The write option is provided to allow conversion to another image format and to allow image property changes (e.g. voxel size, background value) to be saved to disk. The properties are viewed and modified by selecting the "Properties..." button from the menu.
The "Theiler Stage" button allows selection of a Theiler stage reference image and predefined anatomy from the Mouse Atlas CD-ROM. The root of the CD-ROM is defined via the "Theiler setup" option or via an X11 resource.

5.2.2 Anatomy Menu



The anatomy menu becomes available when a Theiler stage is selected from the file menu. The anatomy is read on from the Edinburgh Mouse Atlas CD-ROM and is a menu version of the anatomy hierarchy defined in the database.

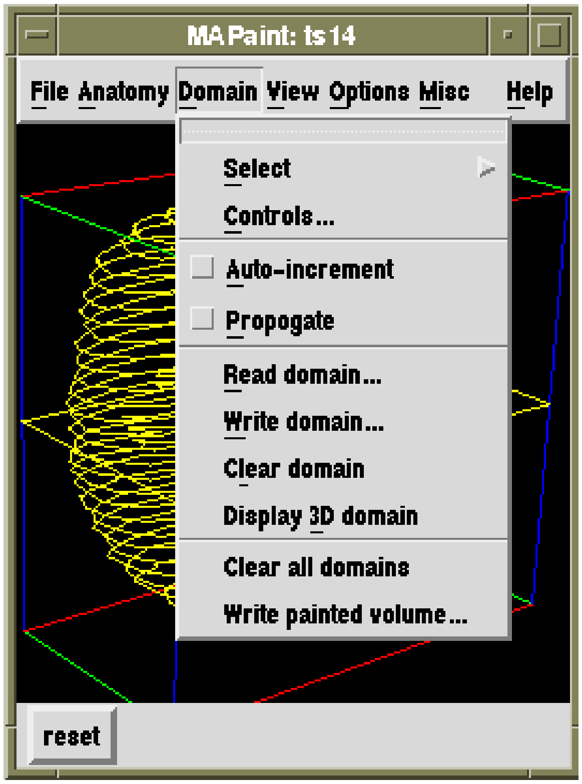
The anatomy database is a standardised anatomy nomenclature arranged as a tree hierarchy. Each anatomical component in the tree represents a subdivision of its parent component with the root of the tree being the entire embryo plus extra-embryonic tissues deemed to be of interest for gene-expression studies in development. Each tissue which is further subdivided shows a “pullright” arrow to the right of the tissue name. Absence of this arrow implies no further subdivisions. Some of the tissues in the tree have been delineated as a 3D domain in the reference image. In this case the tissue name will appear at the top of the list of the pullright menu with the word “domain” appended and it will be separated from component tissues by a line in the menu.

The anatomy list represents all the tissues that are present during the developmental period defined by the stage. The painted domains are those that are recognisable in that particular reference image (which is a snapshot within the range) and therefore will not in general have all tissues present. Furthermore not all tissues are recognisable or distinguishable within the resolution of the atlas and have therefore not been defined. For these reasons not all tissues have pre-defined domains, however every voxel does have an associated tissue.

If the tissue domain button is selected then the corresponding domain will be read in and appended with the currently selected painting domain (see sect 5.2.3. Reading from the anatomy menu also provokes the 3D display of the current domain.

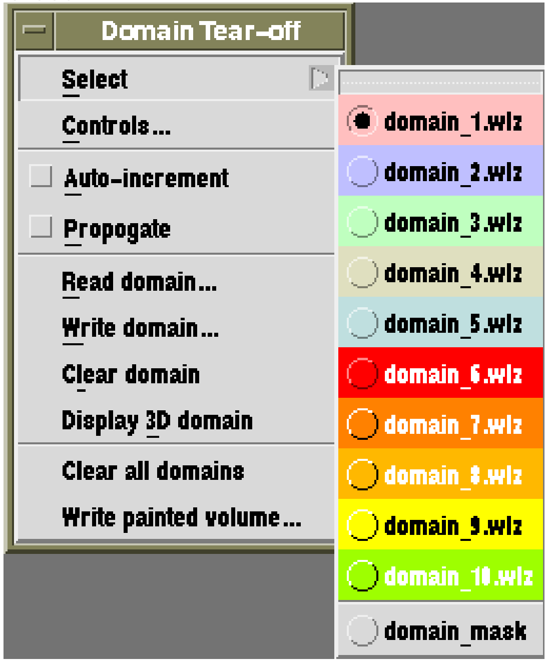
In future versions of the program any component will be selectable with the response that the program will generate the required domain from the given pre-defined domains. In some cases this will involve the union of multiple domains and in others the domain read in will include the requested region (and possibly more).

5.2.3 Domain Menu



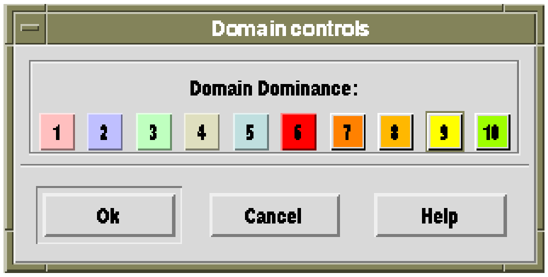
The domain menu allows selection of the "current domain", controls for the dominance ordering, controls for the painting behavior, domain I/O, clear and 3D display. The full "painted volume" can also be saved for use in other 3D display systems (e.g. VTK). Each option is discussed in more detail below.
The domain menu has a tear-off to allow permanent placement of the menu on the screen for convenience.

- Select



This is a pull-right cascade menu which allows the user to select the current domain for any of the domain operations - paint, read, write and clear. If a domain is currently being painted then a new selection for painting will only take effect after that paint session is over. The effect is immediate for the other domain operations. The cascade menu is a set of radio-button toggles in which exactly one is set at a time. The currently selected domain is shown by the state of toggle indicator (by default red). The domain mask is not currently implemented but can be emulated by using the the highest dominance domain as the domain mask.

- Controls...



Selecting "Controls..." will popup the domain controls dialog (above) which can be used to modify the dominance hierarchy. The dominance ordering is left to right on the displayed list as indicated by the numbers. This ordering is also reflected in the "Select" menu, i.e. the domains of higher dominance are higher up the list. To change the dominance ordering select and "drag" the domain to be moved with the middle mouse button and "drop" it on the required location. The domains in the range from the starting position to the drop position will be rotated either up or down in dominance according to the the direction of the drag. The result will be an adjustment of the dominance hierarchy. The "Cancel" button dismisses the controls dialog.

There should be a reset button - to be done!

- **Auto-increment**
  Pressing the auto-increment toggle will enable the auto-increment action after painting the plane. As soon as painting in a plane is completed (right button press or return key after the last painting action) then that view will increment by a distance of one unit and will switch immediately into paint mode. This will continue until either the auto-increment action is switched off (by pressing the toggle again) or the end of the 3D image is reached.

- **Propagate**
  If propagation is enabled by pressing the propagate toggle then the starting paint domain in a particular view will be the union of the existing domain in that view and the previously painted domain. This is of most use in conjunction with auto-increment and means that for each plane the starting paint domain is the correct domain from the adjacent parallel plane. In many cases this will only require minor modification and therefore will be quicker than re-painting the entire domain. If the propagated domain is going to modify an existing domain (of the same colour) on that plane then the user is prompted to check if this is required before continuing:

  Yes: will result in the union of the propagation domain and the existing domain.

  No: will leave the existing domain unchanged. Selecting "No" does not switch propagation off except for this plane.

- **Read Domain**
  Read in a domain object to the currently selected painting domain. A file selection dialog is popped up and the default action is to form the union between the new object and the existing painted regions. Modifying this default will be possible when the control panel is implemented. See Read/Write Domain Dialogs **??** for more details.

- **Write Domain**
  Write out the currently selected domain as a 3D object. A file-selection dialog will be popped-up to allow the user to easily define the image file. Once saved using Write Domain the

filename used will become the default for this domain. See sec-read-write-domains-dialogs for more details.

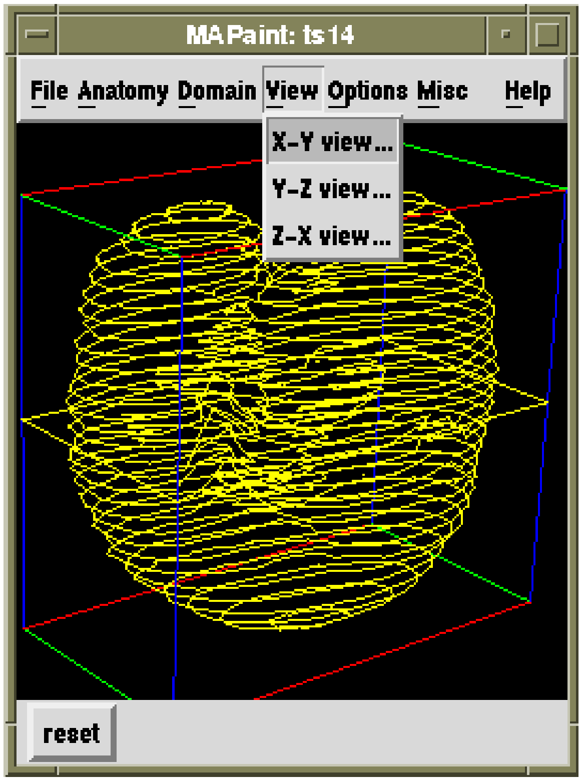
- **Clear Domain**
  Clear the current domain, this could lose a lot of work if used carelessly and therefore the user is asked to confirm the action.

  WARNING!! Until an undo action is implemented, once cleared there is no way of restoring the lost domain.

- **Clear All Domains**
  Clears all domains. Quicker than clearing one at a time but more dangerous. This action erases any memory of defined domains not saved to disc and there is no undo or cancel action. The user is given one chance to quit the option. Under certain circumstances domains erased this way could be recovered from the autosave file.

- **Write Painted Volume**
  This option was put in to allow MAPaint to serve as a means of generating prepared volume images for volume rendering in other 3D visualisation systems e.g. Sunvision, AVS Express, Volvis and VTK. It can also serve as a quick save of all painted domains which can be recovered by reading in the painted volume as an autosave file.

5.2.4 View Menu

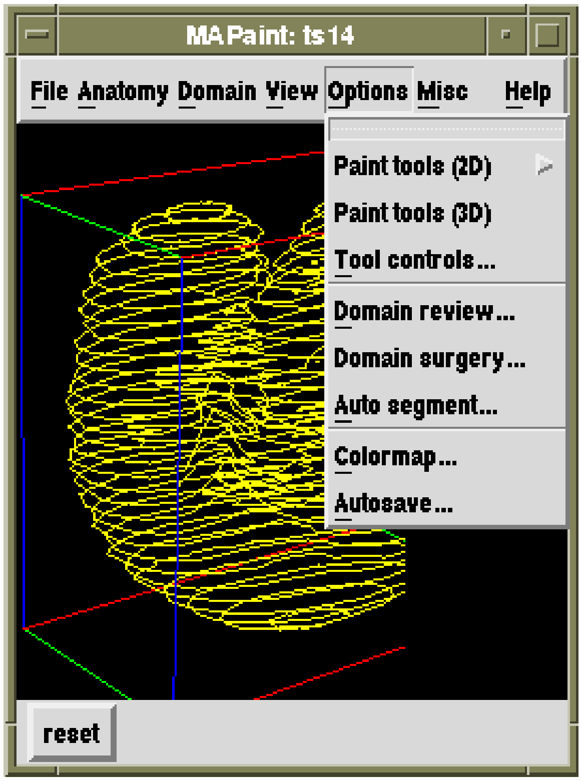


Selecting any of the view-menu entries will pop-up a view dialog showing a cross section of the 3D image and providing a full set of controls for modifying the viewing parameters. For more details see the sec-view-dialog.

The 3D images are defined as a stack of planes with an (x,y,z) co-ordinate frame defined so that the x,y co-ordinates are within the plane and z is perpendicular to the planes. A view of the data is defined by a viewing direction (two angles in 3D), a fixed point in the original co-ordinates and a distance parameter, d. The viewed plane is defined to be perpendicular to the viewing direction and the perpendicular distance from the view-plane to the fixed point is d. For details of how the 3D viewing plane is transformed onto a 2D window see the paint reference document

All the viewing parameters can be adjusted for any view but this menu provides a number of predefined viewing angles. The entries correspond to viewing along the three co-ordinate axes z, x and y respectively however each view can of course be set to arbitrary viewing angles and positions. Full details of the viewing coordinates and the geometric formulae to switch between reference image and section coordinate systems are given in the paint reference document[1].

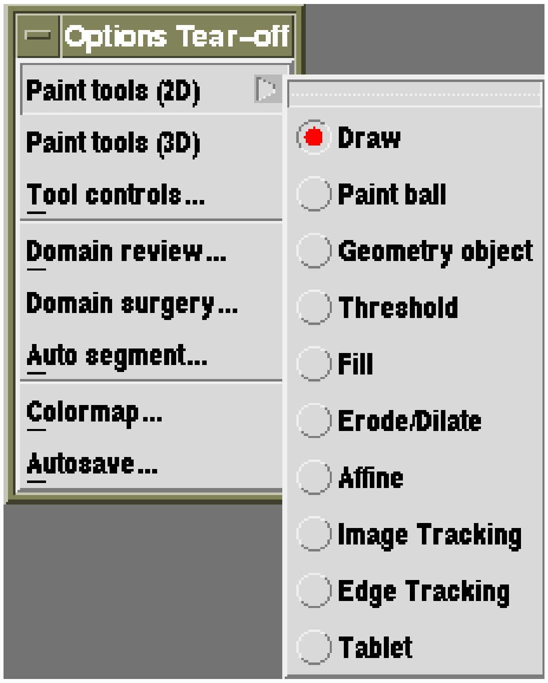
5.2.5 Options Menu



The options menu provides all the methods for painting and otherwise editing the domains. The primary mechanism for defining a domain is by using the 2D tools on a plane-by-plane basis. These are detailed below with further detail given in the the sec-tools-controls-dialog.
A few 3D methods for domain are editing (review and "surgery") as well as options for the display colormap and autosave.

- **PaintTools (2D)**

[1] http://genex.hgu.mrc.ac.uk/Software/paint/paint/paint.html



This is a pull-right cascade menu which allows the user to select the current 2D painting tool. The basic tools are draw and paint-ball which allow the definition are arbitrary 2D domains by drawing around a region which is then filled or by “brushing”. These incorporate both append and delete modes. “Geometry Object” and “Tablet” provide two other primary input modes, the other options all modify or use existing domains.

Details of the settable control parameters for each tool are given in the paint tools controls section. The tools follow a fairly consistent behavior e.g. left-button adds “paint” whereas middle-button or “meta”-left-button removes paint. The currently selected paint-tool is indicated by the appearance of the paint-tool toggle button which will appear depressed and have a red in the indicator. If one of the views has been selected for painting then the cursor in that view window may also have been changed to indicate which tool is selected.

Actions common to all 2D painting tools when the cursor is in the painting view and painting is selected are:

Right arrow: next section - increase distance by 1.

Left arrow: previous section - decrease distance by 1.

Up arrow: next paint tool above on the tool menu list.

Down arrow: next paint tool below on the tool menu list.

<meta>Up arrow: select next domain above on the domain menu list.

<meta>Down arrow: select next domain below on the domain menu list.

Right button: confirm - quit painting in this view. This will result in the domains of this view being installed and provoking all views to update and redisplay. The 3D feedback view will be released.

In all cases except confirm the view will remain in painting mode. If the section has been changed then the domains from the prior section will be committed and marked as changed.

In brief, the 2D paint tools are as follows:

Draw: Use the cursor (by default a dot, but settable to an arrow or sights) to define a closed boundary, as an arbitrary combination of rubber-band straight line links or free-hand curves. On confirm (right-button or return-key) the start and end of the curve will be joined by a straight line, the enclosed region filled and then used either to append to the existing domain (set union) or to subtrat from the existing domain (set difference). If the drawing started with the meta-key (shown with a diamond sign on Sun workstations) pressed, or if the drawing was done with the middle-mouse button then the drawn domain will be subtracted from the existing domain of that colour.

The cursor type is the only settable parameter.

Paint ball: Use the cursor (by default circular, settable to be a square) to paint in or erase an area. Painting will respect the dominance hierarchy, erase will only erase from the domain of the selected colour whatever the dominance.

Settable control parameters are the paint brush/eraser shape and paint brush/eraser size and whether the cursor includes a white border for clarity.

Geometry object: Define the domain by placing a geometric object at the cursor location when the button is pressed. This can be useful for spotting objects for the purposes of location or counting rather than detailed delineation.

Control parameters are the object shape and size.

Threshold: When a pixel is selected by pressing left or middle mouse-button a domain is defined by thresholding the image in a range defined by the grey-value at the selected pixel and the range parameter. This region will comprise all pixels connected to the initial pixel which have value within the range and for which all pixels within the connection path are also within the threshold range (the image is threshold above and below to retain all pixels that lie in the range, this image is then segmented and the region which includes the selected pixel is selected).

On button press the region to be painted will display and if the cursor is dragged right or left the threshold range will increase or decrease respectively. This will have the effect of increasing and decreasing the area of the modified domain respectively.

Settable control parameters are the segmentation connectivity (4- or 8-neighbour) and the initial range parameter. Currently the tool ignores other domains for the purposes of region selection which gives the unexpected result that the region seems to "pass through" domains of higher dominance. This will be changed in future versions.

Fill: This works on the assumption that the view can be segmented into a set of non-overlapping domains. These are defined as connected regions of any pre-defined domains and connected regions of the remaining grey-level image. On selection (pressing the left mouse-button within the region) the region will be filled with the currently selected domain colour if it has higher enough dominance. If delete (meta-button or middle mouse-button) is required only regions of the current painting colour will be deleted irrespective of the dominance.

There are no settable parameters.

Erode/Dilate: Selecting (clicking) any domain with the left mouse-button will dilate it, selecting with middle mouse-button or meta-left button will erode it. This will apply to domains of any colour not just the current selection. Erode and dilate are morphological operations which can be thought of as adding or subtracting all pixels with a certain distance of the existing edge.

The range and distance measure are settable parameters.

Affine: An arbitrary affine transform (translate, rotate, scale and shear) can be applied to the selected domain (any colour). The current transform is selected by clicking the left mouse-button with the control key pressed. The transform is indicated by the cursor shape. The tool controls allow which transforms can be selected, note translate is always selectable and is the default.

The change in position and shape of the selected domain is indicated by the change in shape of the domain boundary which will take effect on button release. Pressing the meta-key when selecting a domain will select all visible regions of that colour for transformation (i.e. select-all).

Image Tracking: Modify the domain propogated from the previous operation (usually the previous section) by using the image information associated with that domain to match to the existing image. The method matches small parts of the image near to the domain and establishes the best match at a series of points along the domain boundary. This is closely related to the "snake" techniques developed for flexible model matching but with a number of differences. The first is that there is no global assumption about the image match cost, the matching at each point on the boundary depends on the image used to define the initial boundary. This breaks the requirement that the snake must have the same appearence at all parts. The second is that there is no constraint on the absolute curvature of the boundary, only a constraint on the change of curvature with respect to the original. This allows the preservation of specific sharp features in the boundary which may be real. There is a control panel which allows the modification of a set of control parameters for the searched area and the snake cost function. The cost function is optimised using Dynamic Programming and is an approximation to the snake cost (making this optimisation globally correct is straightforward and will be implemented soon).

Once the domain has been tracked the tool provide the paint ball behavior to allow immediate editing and correction of the estimated new domain. The parameter settings of the paint ball will be those last set when paint ball was selected (or otherwise the defaults).

Settable parameters are: sampling controls - spacing and range which determine the number of sampling perpendiculars around the boundary and the search distance from the boundary; snake cost parameters, which determine the cost (the new boundary is the minimum cost path) of departing from the original boundary in terms of distance, angle and curvature; image cost parmeters which in this case is the size of the sampled image for comparison (the cost is a normalised sum of the absolute grey-level differences).

Edge Tracking: Modify an existing domain using the same snake as in the image tracking but with the image cost determined by the absolute value of the image gradient (high is low cost) and the continuity of the edge direction so that edges of opposite polarity around the boundary are avoided. The settable width parameter is the full-width half-maximum of the Gaussian function used to smooth the images when the gradient is calculated. This can help to avoid high-frequency noise arising from texture.

Tablet: In many cases it is easier and more convenient to draw on printed pictures of the sections which can be done using a graphics tablet. This is useable simply as a means to control the cursor, the use of a pen to draw is more natural and easier to control than a mouse, but the main use is to draw from the printed sections.

To link tablet coordinates with image coordinates the user defines two fixed points on the tablet which are transformed to two given points within image space. The user first marks each point then draws around the required region. The tool controls allow the user to define the image coordinates of the reference points. These points could of course match fiducial markers within the sectioned material if they exist. The tablet controls are for a Wacom cordless tablet.

- **PaintTools (3D)**
  There are currently no 3D painting tools. In the next version 3D geometric objects, 3D morphological operations and 3D threshold will be available.

- **Tool Controls...**
  The tool controls dialog displays the settable parameters for the currently selected painting tool. Parameters for each tool remain fixed from the last setting irrespective of the current selection.

  See the sec-tools-controls-dialog for details.

- **Domain Review...**
  This will popup the domain review dialog to allow a controlled review of a domain input from disc. Review in this sense means the presentation of the segmented parts of the domain with simple options to the colour destination of each part.

  See the sec-domain-review-dialog for details.

- **Domain Surgery...**
  Domain surgery allows the cutting of existing domains into two parts by defining a cutting surface. Currently the only cutting surface option is the plane defined by a given view. In the next version a “laser” cut will be included.

  See the sec-domain-surgery-dialog for details.

- **Auto segment...**
  Not yet implemented.

  See the sec-autosegment-dialog for details.

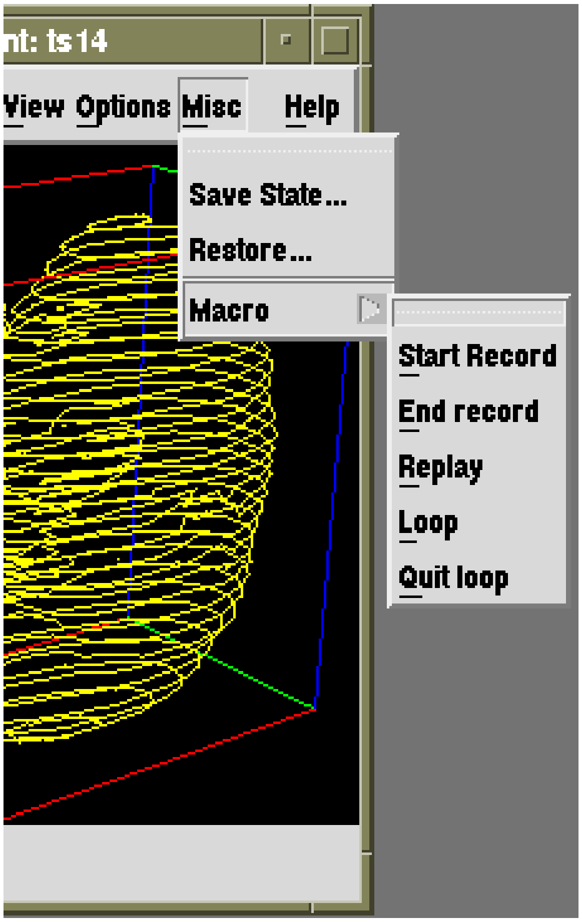
- **Colormap...**
  The display colormap can be controlled by adjusting the visible grey-level ranges and the gamma value. Each domain colour can be also be adjusted as well as the relative contrast of colour to grey-value (wash overlays only).

  See the sec-colormap-dialog for details.

- **Autosave...**
  Like any large program paint is not bug-free therefore an autosave option has been included which saves all of the painted domains at predetermined intervals which defaults to 15 minutes. This time is user settable and autosave can be switched off but is not recommended. Data can be retrieved from an old autosave file and the user is presented with the domains from the saved file and prompted to assign the domains as required.

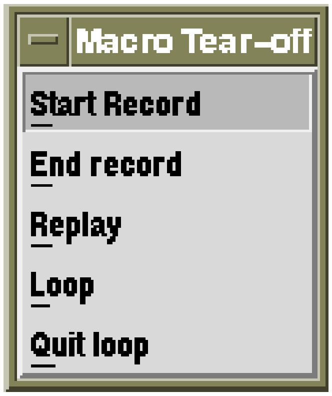
  See the sec-autosave-dialog for details.

5.2.6 Misc Menu



The Miscellaneous menu is the catch all for facilities common to all Mouse Atlas Motif interfaces. Currently this includes the save and restore the programs state and the define and run macro options.
Selecting “save state” will popup a file-selection dialog to allow the current state of the program to be saved to disc in the form of an X-Resources file. These resources can be incorporated as part of a users’ XDefaults and therefore will be invoked as MAPaint starts or can be read in using the Restore option with the result the interface will reconfigure to match the saved state. In this version the view dialog state is not saved and so this option is probably of little use.
The macro system is described below.

5.2.7 Macro Menu



The macro facility allows the user to record a sequence of actions which can then be replayed so that series of repeated actions can be automated. On replay the actions involving interaction with the image e.g. one of the painting tools will wait until painting is confirmed before continuing.

Hint: always tear-off this menu when recording a macro or when running a macro in “Loop” mode. This avoids the menu popup actions being part of the macro and in some circumstances it is very difficult to quit from a loop if the menu is not already visible.

The macro facility records sequences of X11/Xt *actions* which result from user interaction e.g. a button or key press. Sequences of actions can be captured and replayed in order to make repeated sequences more efficient and to help the user avoid spending time and effort on a recordable repeat. The macro menu buttons are:

Start record: Start recording a sequence of actions, all other buttons on this menu except End record will become inactive. Only one macro can be recorded therefore pressing this will automatically delete the previous macro.

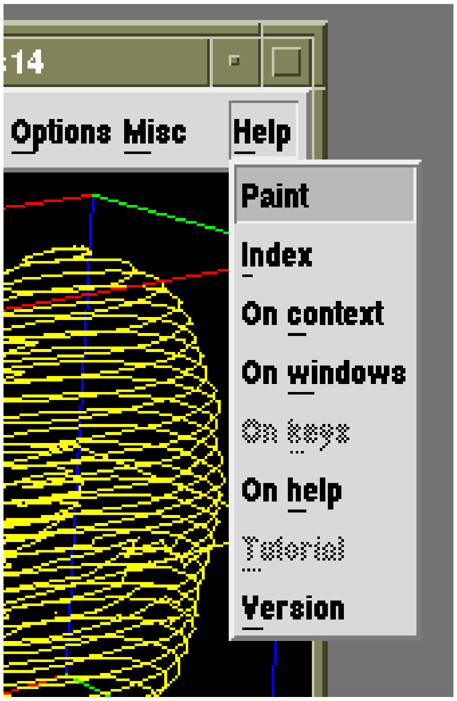
End record: Stop recording the action sequence and make the menu buttons active.

Replay: Replay the action sequence.

Loop: Continous reply of the action sequence. This makes all macro buttons except Quit loop inactive.

Quit loop: Stop the continuous reply of the macro.

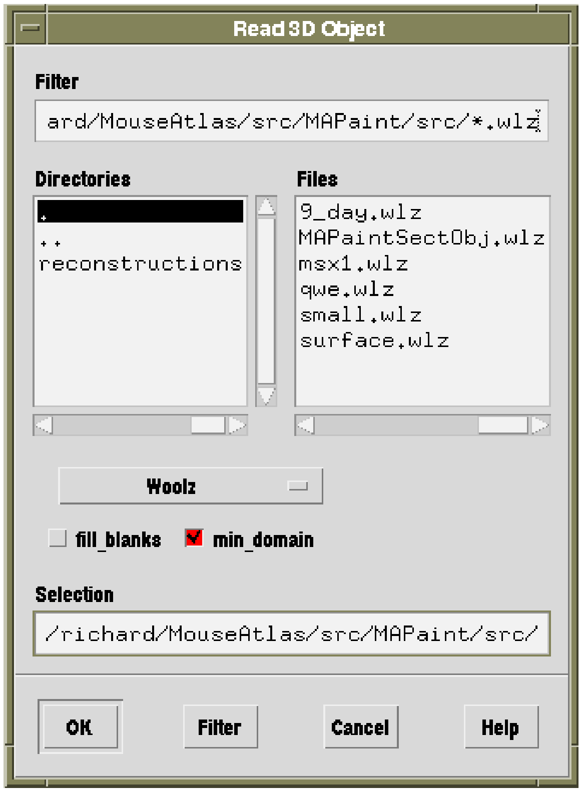
5.2.8 Help Menu



The Help menu conforms to the Motif style providing help "on context" as well as a per-window basis. The Help system requires either that a Netscape browser is running on the X display or that it is possible to start up Netscape on the machine running MAPaint. The help pages will be downloaded from the MRC as html therefore the machine must also be connected to the Internet. The help pages can be installed locally but help "on context" and "on windows" can not be supported without a WWW server.

5.3 Dialogs

5.3.1 Read/Write Reference Dialogs



Filter: ard/MouseAtlas/src/MAPaint/src/*.wlz

Directories:

```
.
..
reconstructions
```

Files:

```
9_day.wlz
MAPaintSectObj.wlz
msx1.wlz
qwe.wlz
small.wlz
surface.wlz
```

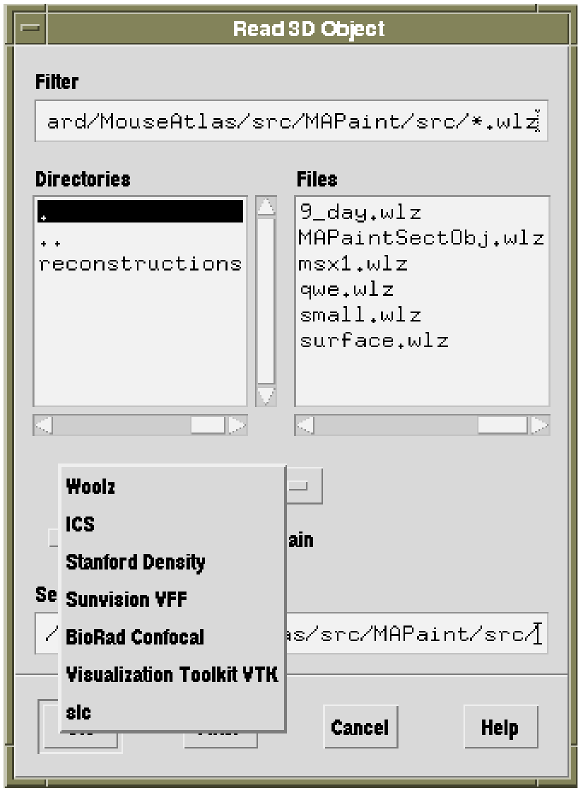
Woolz

☐ fill_blanks ☑ min_domain

Selection: /richard/MouseAtlas/src/MAPaint/src/

When Read Object or Write Object on the File Menu is selected then a file-selection dialog will be displayed. By default the files will be filtered for file extensions of the form .wlz because woolz image files are the default format. Other formats are available from the pull-down menu. A file can be selected by clicking on the file name in the file list widget and the image read by pressing the Ok button. The directory can be changed by changing the directory string in the filter or by selecting a directory in the directory list then pressing OK. Double clicks in either the file or directory list will result in the file being read or the directory changed respectively. For the paint program the woolz object must be of 3D type and must have a voxel table.

In addition to the image format pull-down, the read-file dialog has two toggles to control the behaviour of paint with respect to reference images with blank planes. If ”fill blanks” is selected then then a blank domain with grey-values set to the background value will be inserted for any empty plane of the reference image. The purpose of this is to allow painting in planes which are otherwise not part of the reference image. If ”min_domain” is also selected then paint will introduce a domain to an empty plane which is the union of the two adjacent non-empty planes.



Filter: ard/MouseAtlas/src/MAPaint/src/*.wlz

Directories:
.
..
reconstructions

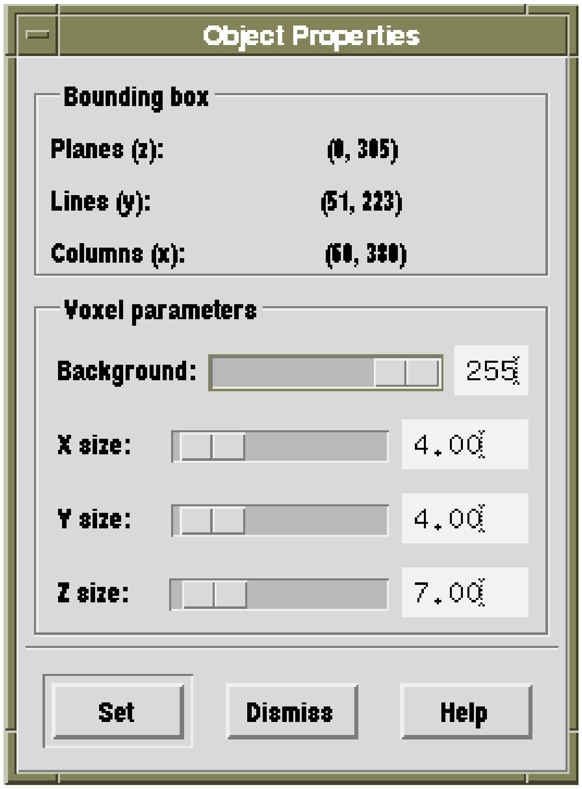
Files:
9_day.wlz
MAPaintSectObj.wlz
msx1.wlz
qwe.wlz
small.wlz
surface.wlz

/ as/src/MAPaint/src/

The choice menu below the directory and file windows alows the selection of other image formats. Selecting a format will automatically reset the filter for the default file extension. If the required file does not have this extension then it can still be input but to make the file visible in the file list change the filter line to read in all files by removing the .??? following the arterisk (*) and pressing "return" or click the filter button.
See sec-input-data-formats for a list of the available formats.

The sec-output-data-formats include the option of writing to a series of 2D image files.

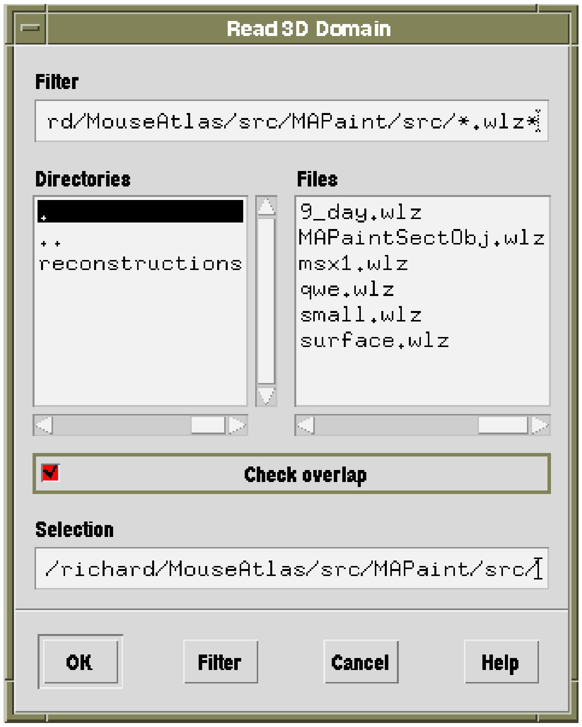
5.3.2 Reference Properties Dialog



Planes (z): (0, 305)
Lines (y): (51, 223)
Columns (x): (60, 380)
Background: 255
X size: 4.00
Y size: 4.00
Z size: 7.00

The properties dialog provides information about the reference image and allows a number of parameters to be set. The background property affects the way empty regions of the reference image are displayed and the voxel sizes X-, Y-, and Z-size will change the voxel-size property. This will not affect the displayed reference image in this version of paint but will make a difference to the saved domains when viewed in for example AVS. If the changed properties are required to be set in the reference image then the reference image must be written to disc.

5.3.3 Read/Write Domain Dialogs



Filter: rd/MouseAtlas/src/MAPaint/src/*.wlz*

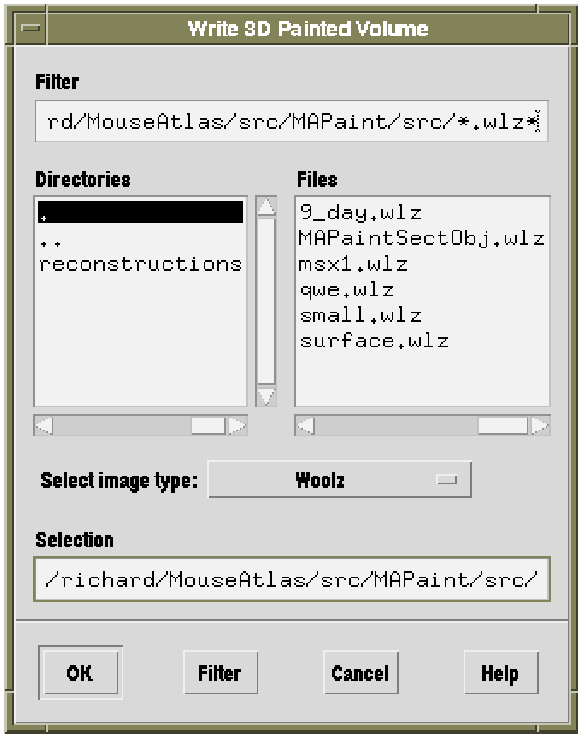
Directories:
.
..
reconstructions

Files:
9_day.wlz
MAPaintSectObj.wlz
msx1.wlz
qwe.wlz
small.wlz
surface.wlz

☑ Check overlap

Selection: /richard/MouseAtlas/src/MAPaint/src/

5.3.4 Write Painted Volume Dialog



Filter: rd/MouseAtlas/src/MAPaint/src/*.wlz*

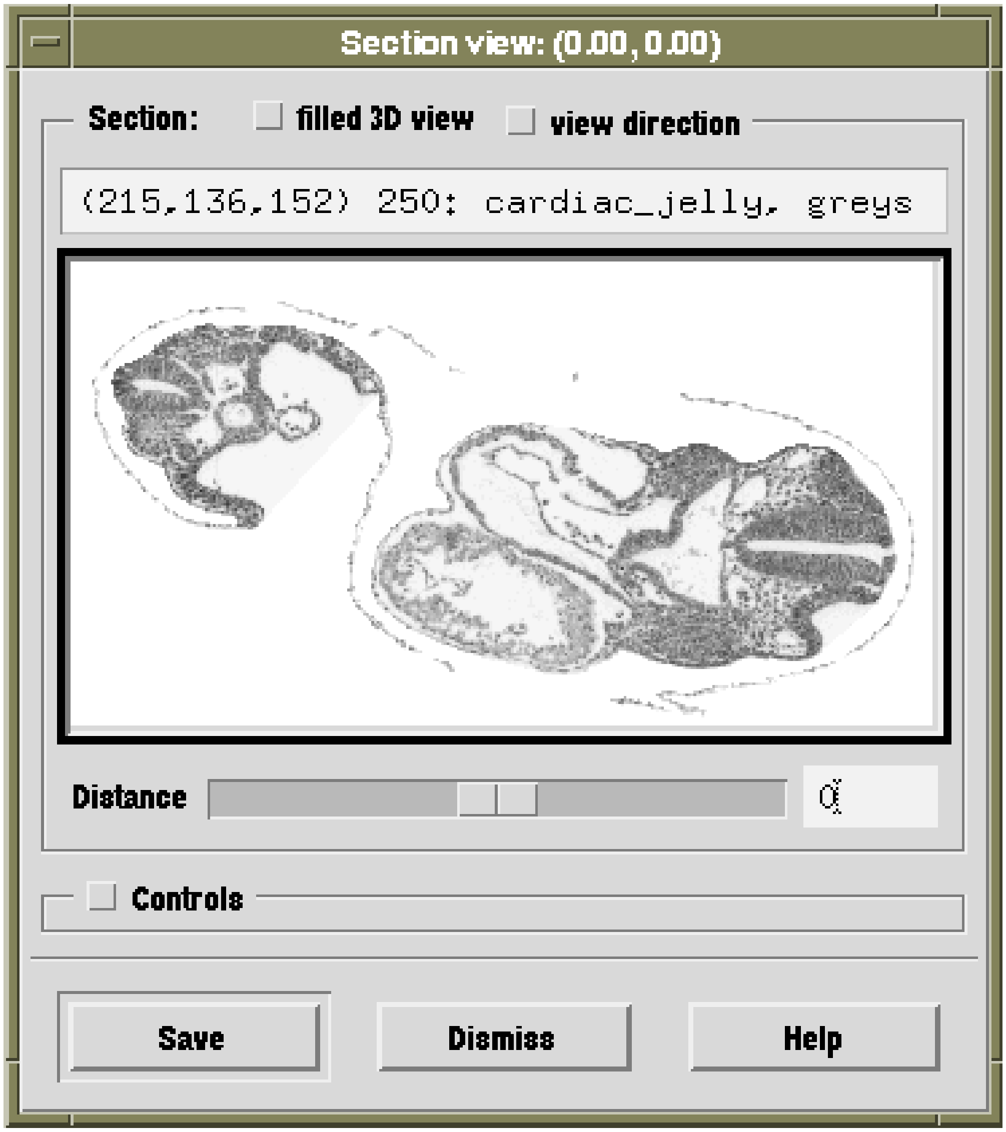
Directories:
- .
- ..
- reconstructions

Files:
- 9_day.wlz
- MAPaintSectObj.wlz
- msx1.wlz
- qwe.wlz
- small.wlz
- surface.wlz

Select image type: Woolz

Selection: /richard/MouseAtlas/src/MAPaint/src/

5.3.5 View Dialog


Section view: (0.00, 0.00)

(215,136,152) 250: cardiac_jelly, greys

Distance 0

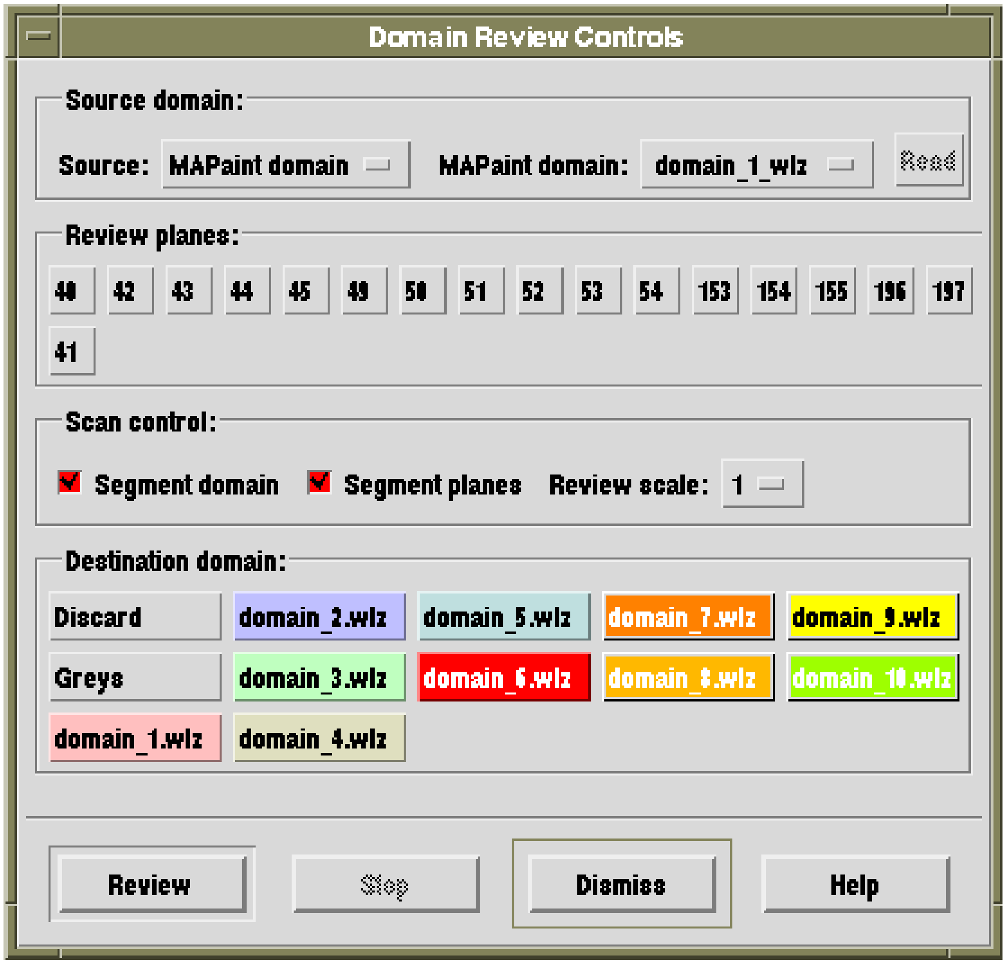
The view dialog shows a 2D section through the 3D image data. The section displayed is defined by the viewing parameters (see the paint reference document). The distance parameter is accessible directly via the slider on the view dialog and is defined as the perpendicular distance from the fixed point to the displayed section. The slider range will be set to allow the display to pan through the whole 3D image. When the distance is zero the fixed point is in the plane and will be visible if the fixed point display toggle has been set (see fixed-point menu below).

The controls toggle will make the section orientation controls visible which allows a number of modes of navigation through the 3D volume.

The basic mechanism for painting is the user selects a particular view for defining the domain. Once that view is selected the window border will be inset and coloured red. All controls to that window will be disabled and no other view can be selected until the painting operation in the selected view is complete. Completion of a painting operation is usually done by pressing the right mouse button or . The paint operation that will be invoked is the tool selected in the Options - Paint Tools menu.

5.3.6 Tools Controls Dialog

5.3.7 Domain Review Dialog



Domain Review Controls

Source domain:

Source: MAPaint domain MAPaint domain: domain_1_wlz

Review planes:

40 42 43 44 45 49 50 51 52 53 54 153 154 155 196 197 41

Scan control:

☑ Segment domain ☑ Segment planes Review scale: 1

Destination domain:

The purpose of domain review is to allow the user to scan through the component parts of an input domain allocating as required to different paint domains, including clear to grey and ignore (i.e. leave existing domains alone). The dialog is divided into four regions plus the action button bar at the bottom. The four regions are:

Source domain: use the controls to select the source of the domain for review. The *Source* choice menu allows selection between *Review domain*, *MAPaint domain* and *Read domain*. When new domains are read in from the Read domain option of the Domain menu the new domain is checked against existing painted domains. If there is overlap then the user can choose for the intersection region to be installed as the *Review domain* for later review. This will be used as input to the review process if the Source selection is *Review domain*.

The MAPaint domain choice means tha an existing paint domain, as determined by the *MAPaint domain* choice box can be used as input for review. Finally the Read domain option activvates the Read button to allow a domain (2D or 3D) to be read from disk.

Review planes: Once a domain has been installed for review, if it is a 3D object then the planes of that object will be listed within the Review planes control area. This is to allow the user to select specific planes to review. The process of review will automatically loop through all planes and all parts within a plane and when all parts within a plane are assigned the button for that plane becomes insensitive.

Scan control: Set the *Segment domain* toggle on if the domain on each plane should be segmented into unconnected regions for independent allocation. Set the *Segment planes* toggle on if the planes of the input domain should considered independently (this toggle has no action in version 1.0). When domains are being reviewed the controls of the section view used for review are insensitive, use the *Review scale* choice menu to adjust the display scale.

Destination domain: When review is in progress the region to be assigned is flashed on the section view. The user can then choose to which domain the flashing region should be appended by pressing the corresponding button in the Destination domain area. Discard means that the flashing region will ignored and any domains within that region left unchanged. Greys means that the flashing region will be set to Grey thereby clearing any parts of any domain within the region. Note assignments respect the dominance hierarchy.

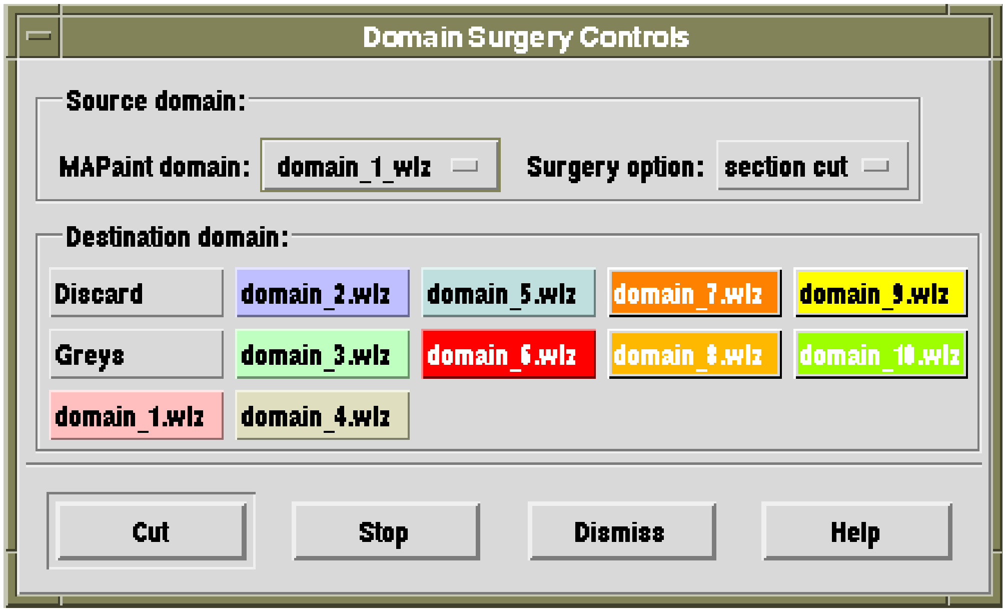
If no assignment is possible but it is required that the region is kept for further review then press the *Stop* button or select a different plane by pressing the corresponding button.

Action buttons:

- **Review:** Start the review process. This will loop through all planes and all parts within each plane (if Segment domain is selected) offering each part for review by flashing the region on the section view window. If a new plane is selected then the review will continue from that plane onwards. Pressing Review will popup the required section view.
- **Stop:** Stop the review process and popdown the section view.
- **Dismiss:** Stop the review process and popdown the Domain review dialog.
- **Help:** Probably does nothing.

If the source domain read from disk is a 2D image then it is assumed that the user wishes to map this domain onto an arbitrary section through the reference image. In this case the user is prompted to set the view section as required then can continue to review in the usual way.

5.3.8 Domain Surgery Dialog



Domain Surgery Controls

Source domain:

MAPaint domain: domain_1_wlz

Surgery option: section cut

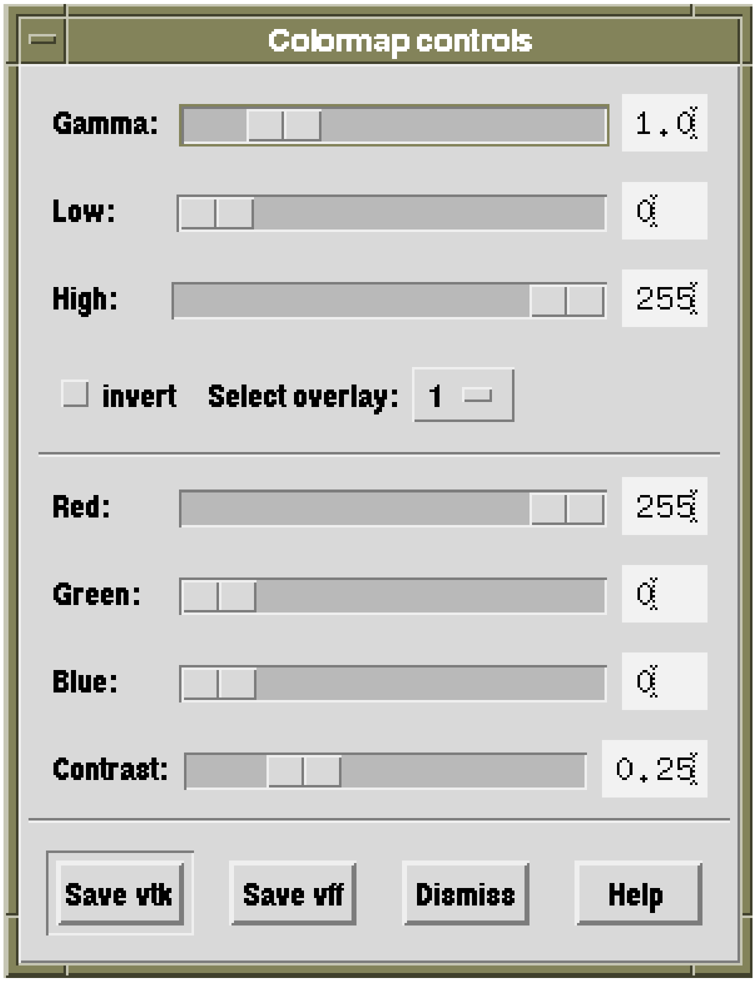
The basic idea with domain surgery is that the user defines a source domain and a method of cutting that domain in 3D. The user is then presented in the 3D feedback window with a flashing display and can assign that region as required using the Destination domain buttons.

Currently the only Surgery option is to use a section plane determined by a view dialog. For this option the user adjusts the view controls of a section view until it corresponds to the required cutting plane. That view should then be selected for painting and then the Cut button is pressed. The selected domain, as determined by the Source domain choice, is then spilt into two parts, one on either side of the cutting plane and presented in turn to the user as flashing regions in the 3D display. The user can then select the destination domain for the flashing region including reset to grey and discard which leaves that part of the domain unchanged.

5.3.9 Autosegment Dialog

Not yet implemented.

5.3.10 Colormap Dialog



Gamma: 1.0

Low: 0

High: 255

□ invert Select overlay: 1

Red: 255

Green: 0

Blue: 0

Contrast: 0.25

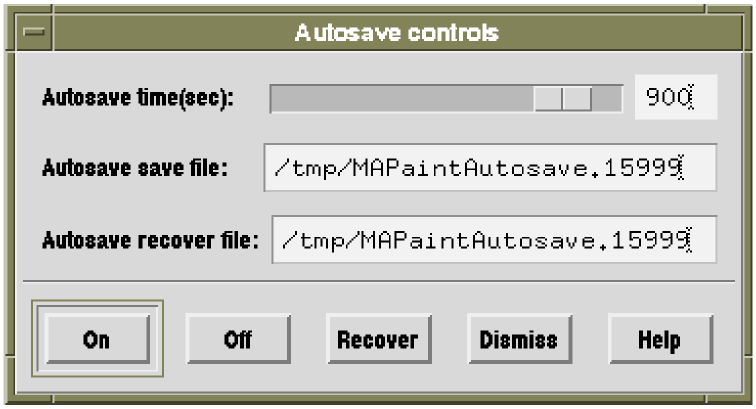
The colormap controls allow some manipulation of the display of the grey-level image and midification of the colours for each overlay. In the case of the wash overlays there is also control of the relative contrast of the grey to colour. A contrast value of 1.0 implies solid colour, a value of 0.0 implies no colour.

The controls *Gamma*, *Low*, *high* and *invert* allow modification of the grey-value look-up table. Image grey-values within the range [low, high] will be mapped onto the display values [0,255] with lower values mapped to zero and higher values to 255. The mapping will use the value of gamma set which by default is linear (gamma=1.0). The look-up table can be inverted by selecting the invert toggle. None of these operation actually modify the grey-level image, just how they are displayed.

The *select overlay* choice menu defines which overlay the colour and contrast controls will modify. The colour sliders control the colour mixture for each domain and the contrast will modify the relative "transparency" of the wash overlay.

The buttons *Save vtk* and *Save vff* enable the colormap to be saved in a suitable format to be read into other programs in conjunction with the painted volume.

5.3.11 Autosave Dialog



Autosave time(sec): 900

Autosave save file: /tmp/MAPaintAutosave.15999

Autosave recover file: /tmp/MAPaintAutosave.15999

MAPaint occasionally crashes (what a surprise). To avoid losing too much work an autosave mechanism has been implemented which will save the entire painted volume at regular intervals. This painted volume can then be read in and the domains extracted and assigned as required. The original domain names are not currently restored so it is up to the user to be able to tell which is which.

In this version the restore mechanism is very slow so patience is required when recovering domains from the autosave file. Each control is settable via an X11 resource. By default the autosave directory is /tmp which on Solaris is memory mapped therefore the autosave operation is usually very fast. If the file has to be written to disk then the process may take a number of seconds depending on the size of the reference image.

The autosave filename will by default have the process ID appended in order to generate an unique filename. The button controls are:

On: switch the autosave mechanism on.

Off: switch the autosave off.

Recover: recover domains from the given Autosave recover file. It is usually worth switching autosave off for this action and in general any existing domains should have been saved. If the process runs out of memory during this action the program will usually quit.

Dismiss: popdown the Autosave dialog.

Help: probably does nothing.

6 Using a Graphics Tablet

MAPaint is designed and intended for use as a tool to define arbitrary regions within the 3D space defined by a grey-level voxel image. The painting actions are all undertaken using the screen pointing device which is typically a mouse or trackball. For drawing accurately this can be difficult and many users find it more natural to use a pen like device for drawing. The simplets device for this is a graphics tablet and therefore the possibility of using a tablet has been included. For this purpose support for the WACOM UltraPad tablet has been programmed in, specifically with code for the cordless pen and 4-button puck.

The tablet can be used in two ways. The first is an alternative input to control the screen cursor. In this mode the pen or puck can be used as a simple alternative to using the computer mouse and the benefit is that drawing with a pen is often more natural and easier to control for detailed positioning. The user is still required to look at the screen to define the domain. To use the system in this mode the sofware that comes with the tablet for the host machine is installed as per the instructions and no modifications are required in MAPaint (in this mode,so far as the program is concerned, user-events generated by the tablet are indistinguishable from that of the computer mouse).

In the second mode of use, printed images of each section are generated and the user draws with the tablet pen or puck directly on the sheet. The tablet coordinates generated by this action are then converted directly to voxel coordinates and thereby to define the domain. For this mechanism the tablet coordinates are read directly by MAPaint and converted using the coordinate mapping defined by the tablet reference points set in the tool controls dialog. The tool controls points are the coordinates in the reference image of two points that will be input from the tablet. Typically these will correspond to some fiducial point printed with the image on the sheet but could also be fiducial markers embedded with the embryo and stained to be visible.

6.1 Installing a WACOM tablet

MAPaint will automatically detect and initialise the tablet if it is switched on and connected to serial port A. If it fails to find the tablet then error messages will appear on the command line when an attempt is made to use the tablet option.

6.2 Generating Printed Views

6.3 Calibrating the Coordinate Mapping

7 Painting Techniques and Strategies

8 Conditions of Use

The program MAPaint is provided in binary form, freely for use by non-profit institutions. There is no charge for the use of the software and no guarantee. The program is used entirely at the users own risk and the author and Medical Research Council can not be held responsible for any harm, real or perceived, arising from its use.

If versions are required for other Unix operating system versions or machine architectures please contact the author and source code can be provided for that purpose provided all new versions can similarly be made freely available.

9 Version

MAPaint - an X11/Motif program to interactively segment a 3D voxel image.

This program has been developed at the Medical Research Council (UK) Human Genetics Unit as part of the Mouse Atlas and Gene Expression Database project.

You are using version 1.0. For more information contact:

Author:	Richard Baldock	MRC Human Genetics Unit
Internet:	Richard.Baldock@hgu.mrc.ac.uk	Western General Hospital
Phone:	+44-(0)31-332-2471	Crewe Road
FAX:	+44-(0)31-343-2620	Edinburgh, EH4 2XU, UK.