

vpCurrentMatrix(3)

vpCurrentMatrix(3)

NAME

vpCurrentMatrix – choose the current transformation matrix

SYNOPSIS

#include <volpack.h>

vpResult

vpCurrentMatrix(vpc, option)

vpContext *vpc;

int option;

ARGUMENTS*vpc* VolPack context from **vpCreateContext**.*option* A code specifying one of the VolPack transformation matrices (VP_MODEL, VP_VIEW or VP_PROJECT).**DESCRIPTION****vpCurrentMatrix** is used to choose the current transformation matrix. The *option* parameter must be one of the following: VP_MODEL modeling transformation

VP_VIEW viewing transformation

VP_PROJECT projection transformation Subsequent calls to the matrix transformation functions (**vpIdentityMatrix**, **vpTranslate**, **vpRotate**, **vpScale**, **vpMultMatrix** and **vpSetMatrix**) are applied to the new current transformation matrix.

Typically the modeling transformation is used to specify a transformation from object coordinates to world coordinates. In object coordinates the volume data is located inside a unit cube centered at the origin (corners at (-0.5, -0.5, -0.5) and (0.5, 0.5, 0.5)). The default modeling transformation is the identity, so the volume is initially located inside the unit cube in world coordinates. By default, the current matrix is VP_MODEL. The modeling transformation is also applied to light direction vectors (see **vpSetLight**(3)).

The viewing transformation is used for transforming world coordinates into eye coordinates. In eye coordinates the viewer is looking down the Z axis in the negative direction. The default value of the matrix is also the identity, so the viewer is looking down the Z axis in world coordinates as well. Typically this transformation matrix is used to specify the location and direction of the viewer. It is also possible to combine the modeling and viewing matrices (as in the OpenGL viewing model), in which case you can store the composite transformation in either the model or the view matrix set the other matrix to the identity.

The projection transformation specifies the transformation from eye coordinates to clip coordinates. In the clip coordinate system the entire viewing frustum lies inside a cube defined by the planes X=1, X=-1, Y=1, Y=-1, Z=1 and Z=-1. This transformation also specifies the type of projection used to produce a 2D image. The 2D image is computed by projecting along the Z axis in the clip coordinate system and then scaling in the X and Y dimensions to the final image size. The routines **vpWindow** and **vpWindowPHIGS** are useful for computing projection transformations.

STATE VARIABLES

The current transformation matrix code can be retrieved with the following state variable codes (see **vpGeti**(3)): VP_CURRENT_MATRIX.

ERRORS

The normal return value is VP_OK. The following error return value is possible:

VPERROR_BAD_OPTION

The *option* argument is invalid.**SEE ALSO**

VolPack(3), vpCreateContext(3), vpGetMatrix(3)

