VPascal 5.0 Automation Language

VPascal is a powerful programming language that allows you to easily automate complex or repetitive tasks in V++. It is a full featured language which implicitly supports image, vector and matrix math enabling you to calculate complicated image expressions in a single line of code. You can handle images as easily as other languages handle individual numbers. VPascal is easy to learn because it uses virtually the same syntax as the popular Pascal programming language (widely used worldwide as a teaching language) but has many enhancements for imaging applications.

Module Recorder

VPascal programs are referred to as "modules". The module recorder allows you to create simple modules by pointing and clicking - without writing a single line of code yourself. This is a great way to speed up development or to learn about VPascal programming.

Mathematical Expressions

You can enter equations to perform image processing operations just as you would write them down mathematically. For example, a reference subtraction requires a line of code like this:

```
NewImg := OldImg - RefImg ;
```

No longer do you have to write code to loop over every pixel in the image - the automation language takes care of that for you.

When you do need to get at individual pixels, the image is accessed just like a regular array:

```
x := NewImg[ 23,54 ] ;
```

VPascal's powerful image range notation enables you to refer to whole regions within an image in a single expression. For example, the following code copies the third and fourth columns to the first and second columns:

```
Image[..,0..1] := Image[..,2..3] ;
```

Expressions can be of any complexity, and you can freely mix scalars (single numbers) and images. For example, a simple flat-field calculation might be:

```
New := 123*( Old - Ref )/( Flat - Ref ) ;
```

Sequences and Stacks

A sequence (or stack) is the natural extension of an image into the third dimension. Sequences are treated just like 3D arrays of numbers so to get at a single pixel just use three coordinates.

```
APixel := MySeq[ 73,54,123 ] ;
```

You can even refer to entire 3D regions within a sequence using a single line of code, thanks to index range notation:

```
Volume := MySeq[ 23..50, 54..102, .. ] ;
```

In addition, numerous built-in functions are provided for getting at any slice, sub-sequence or line in a sequence.

Flow Control

All the usual programming elements are included to control the flow of a module: if-then-else, repeat-loops and while-loops. For example, to avoid a divide-by-zero error in a flat-field correction use the if-then-else statement:

```
if Any( Flat - Ref = 0 ) then
  WriteError( 'Divide by zero!' )
else
  New := 123*( Old - Ref )/( Flat - Ref ) ;
```

Toolbar and Menu Activation

Any VPascal module can create toolbar buttons and menu commands which activate the functions it has defined. Modules can be pre-compiled and ready to go (and even executed if required) when V++ starts.

Built-in Functions

VPascal has more than 600 built-in functions to handle image processing operations such as filtering, zooming, histogram calculation, object analysis, morphology, and much more besides. Numerous utility routines take care of other tasks.

Spreadsheet Control

V++ includes a built-in Excel-compatible spreadsheet with formula calculation and styles. VPascal includes a set of functions to create and manipulate spreadsheets, enabling you to output results in tabular format and save them in several Excel compatible formats (including XLS). Your VPascal code can even insert or modify formulas on the fly.

Chord Tables

Chord tables are a powerful and compact data structure that allow you to address complex shaped regions within an image all at once. VPascal is equipped with functions that allow your code to easily create chord tables and use them to manipulate image data.

Dynamic Data Exchange (DDE)

Use DDE to control other applications or have them control V++, even over a network. You can not only share an entire VPascal program via DDE but even share the VPascal program’s variables and individual procedures! This lets you link variables, results and images directly to applications like Microsoft Excel.

Serial Port Control

Full control of serial ports is built into VPascal. This makes it easy to control external serial devices such as shutters, stage controllers and photometers.

Built-in Dialogs

There are several built-in dialog boxes to prompt the user for input, such as entering a number or a string, or selecting an image from the V++ desktop. For example, the statements:

```
    Msg := 'Choose an image to transform' ;
    SelectImage( Msg,MyImg ) ;
```

display a dialog box with the message, together with a list of all the desktop images, and returns the selected image in the variable MyImg.

User-defined Functions

Automation modules are easier to understand and maintain if they are compact and modular. Functions and procedures are sections of code that neatly encapsulate the complexity of an algorithm, and can be invoked just like any of the built-in functions.

External Functions

Incorporating special-purpose code you have written yourself, or library functions you have purchased, is incredibly easy. Simply declare the functions in your module and use the external keyword to indicate that they can be found in a particular DLL. Then you can just call the external functions the same way as any function defined in VPascal itself. You can even call the Windows API directly, giving a VPascal module an extremely high level of control.

Windows Interaction

Controlling the appearance of many on-screen images is simplified using the built-in window management functions. Images are easily tiled or cascaded, or the size and position can be set precisely. During lengthy operations keep the user informed of progress by updating the status bar with informative messages.

Extensive On-line Help

Comprehensive help is available for beginners and experts alike. First-time users can follow the step by step instructions on how to write, compile and run a module. The many built-in functions have detailed descriptions of parameters, return values and - most importantly - example code you can cut and paste into your own modules.

VPascal Language Elements

Data Types
Scalars (boolean, 32-bit integer, 64-bit floating-point, 128-bit complex, 48-bit RGB, 96-bit floating-point RGB), images & arrays (1-bit, 8-bit, 16-bit signed & unsigned, 32-bit signed, single- & double-precision floating-point, single- & double-precision complex, 24-bit RGB, 48-bit RGB & 96-bit RGB), strings and the null type (uninitialized). Arrays may be 1, 2 or 3 dimensional.

Declarations
Constant declarations and variable declarations. User-defined procedure and function declarations with local and parameter declarations. Declare variables or procedures as DDE "shares". Declare procedures or entire programs as button or menu-activated. Declare external DLL library functions.

Control Statements
Program flow control with if-then and if-then-else statements. For-loops for executing statements a predefined number of times. Repeat-loops for executing statements until a condition is true. While-loops for executing statements while a condition is true.

Expressions
Mathematical expressions of any complexity. Expression operators are arithmetic, logical, or relational. Expression operands are scalars, images and function results. Full support for complex and RGB numbers. Expressions may involve scalars, vectors, matrices, images, sequences and combinations of these. Image expressions are treated identically to scalar expressions.

Array Index Notation
Arrays and images are formally identical. Arrays may be indexed using scalars or scalar expressions. Image sub-regions and sub-volumes may be addressed directly using index range notation. Range checking occurs at runtime.

Arithmetic Operators
Addition (+), subtraction (-), multiplication (*) division (/) and modulo. Arithmetic operations are defined for all scalar-scalar, image-scalar and image-image operands. Floating-point to fixed-point conversion using rounding, truncation, ceiling or floor functions.

Logical Operators
Bit-wise and boolean logical operators: AND, OR, XOR, NOT, NAND, NOR and XNOR. Logical operations are defined for all fixed-point scalar-scalar, image-scalar and image-image operands. Bit-wise shift operators: SHL, SHR and SAR. Bit-wise rotate operators: ROL and ROR. Shift and rotate operations are defined for all fixed-point scalar-scalar and image-scalar operands.

Relational Operators
Greater than (>), less than (<), equal to (=), greater than or equal to (>=), less than or equal to (<=) and not equal to (<>). Relational operations are defined for all scalar-scalar, image-scalar and image-image operands.

User-defined Routines
User-defined procedures and functions. Pass parameters (scalars, images and strings) to routines. Functions can return any type, including images. Local constant and variable declarations. Late-binding of parameters and function results provides for polymorphic code.

Parameter Passing
Parameters are passed to procedures and functions by value (a copy is made prior to entry) or by reference (address of parameter is passed).

Predefined Dialogs
Utility dialogs to simplify user interaction: error dialog, information dialog, confirmation dialog, image selection dialog, string-selection dialog, number entry dialog, string entry dialog and halt dialog.

External Code
Call external DLL functions written in any language. Call Windows API functions directly. Execute external applications. Add support for specialized hardware.

Windows Interaction
Control size and position of image windows on desktop. Show and hide image windows. Minimize, maximize and restore image icons and V for Windows application. Tile or cascade desktop windows and arrange desktop icons. Print. Write to the status bar. Create custom toolbars and menus. Call Windows API functions directly.

Built-in Functions
Over 600 standard functions built-in. Categories include: image and spectrum data handling, array management functions, chord table management, color functions, data type conversion, text editor management, spreadsheet management, file management, convolution and order-statistic filters, geometric functions, image management, desktop-image management, user input & output, intensity (contrast and histogram) functions, math functions, memory management functions, object analysis and counting, graph plotting, image and text printing, serial communications, DDE (dynamic data exchange client and server), statistics (regular and ensemble), string functions, XML attachment management, utilities (e.g. timing functions, playing sounds), camera control, video capture, TWAIN devices, spectrometer control (with optional RCubed plug-in) Windows interaction and much more.