

Wykład 6

12.04.2024 r.

Grafika

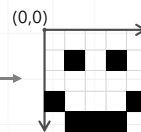
Adrian Kaim



Politechnika Wrocławska

Graficzne typy danych

LabVIEW korzysta z grafiki rastrowej, opartej na układzie kartezjańskim o początku w lewym górnym rogu



Typ Picture

new picture

Podstawowy typ grafiki, pozwala na wyświetlanie i kontrolę grafik w LabVIEW.

Typ Image Data

image data

image type
image depth
image
mask
colors
Rectangle

Klaster, który oprócz samych pikseli tworzących obraz, zawiera inne dane np.: głębia barw, paleta kolorów, przezroczystość. Najczęściej używany do obsługi zewnętrznych plików.

Tablica 2D

24-bit pixmap

*2F68AB	*3069AC	*336AA1	*346BAE	*326BB0
*316AA1	*316AA1	*336AA1	*366DB0	*346DB2
*336CB1	*346DB2	*366DB0	*366DB0	*376EB1
*346DB2	*356EB3	*376EB1	*386FB2	*386FB2
*336EB0	*3770B3	*3970B3	*3872B5	*3973B3
*3570B2	*3770B3	*3A71B4	*3872B5	*3973B3
*3770B3	*3770B3	*3872B5	*3872B5	*3A74B4
*366FB2	*3770B3	*3872B5	*3C73B6	*3E75B8
*3770B5	*3972B7	*3C72B8	*3C73B6	*3E75B8
*3972B7	*3972B7	*3B71B7	*3C72B8	*3D73B9
*3A73B8	*3A73B8	*3D73BE	*3E74BA	*3C75B7

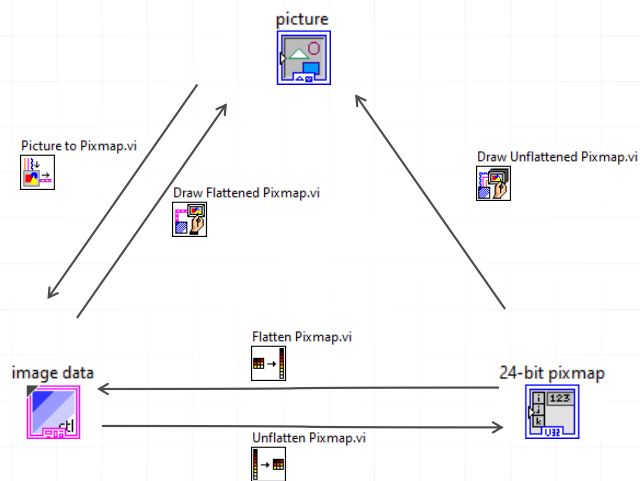
Traktuje grafikę jak zwykłą tablicę 2D. Daje dostęp do wartości każdego piksela.

W LabVIEW obsługiwać można obrazy w postaci rastrowej, opartej na kartezjańskim układzie współrzędnych. Pierwszy piksel znajduje się w lewym górnym rogu i ma współrzędne (0, 0), a kolejne wiersze i kolumny (indeksy) naliczane są odpowiednio w dół i w prawo. Zazwyczaj informacje o kolorze piksela przechowywane są w postaci 24 bitowej, której pierwszy bajt opisuje intensywność barwy czerwonej, drugi zielonej, a trzeci niebieskiej. Możliwe jest także wykorzystanie mniejszej przestrzeni barw (8-, 4- i 1-bitowej), ale wymaga to zdefiniowania palety dostępnych kolorów.

W LabVIEW dostępne są trzy typy danych odpowiadające za przechowywanie grafiki:

- *Picture* – najbardziej podstawowy typ, który pozwala na wyświetlanie i kontrolę grafik w kodzie. Większość funkcji pozwalających na manipulację obrazem korzysta z tego typu.
- *Image Data* – złożony typ danych, który w postaci klastra przechowuje obraz wraz z podstawowymi metadanymi.
- Tablica 2D – pozwala traktować grafikę jako zwykłą tablicę w reprezentacji 24-, 8-, 4- i 1-bitowej oraz umożliwia odczyt i manipulację wartościami poszczególnych pikseli.

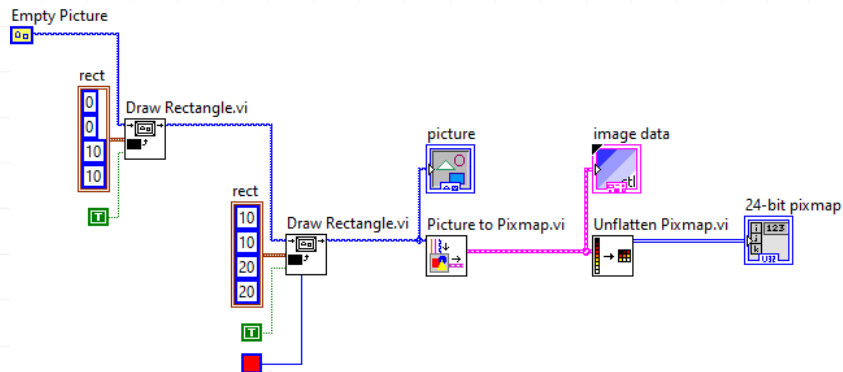
Konwersja typów graficznych



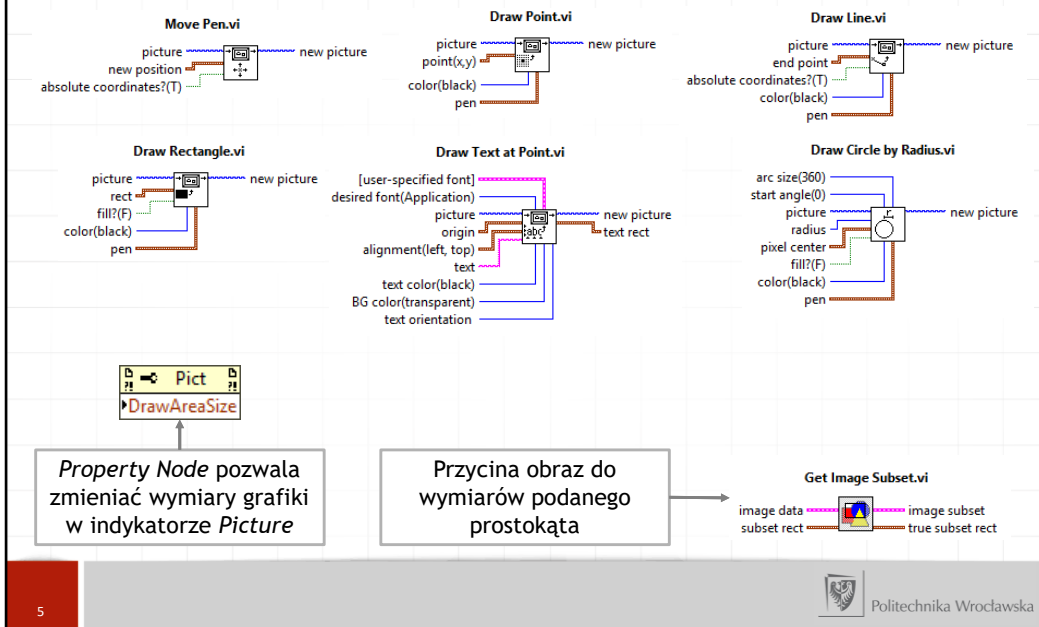
Przechodzenie między typami umożliwia zestaw funkcji do ich konwersji:

- *Draw Unflattened Pixmap.vi*
- *Flatten Pixmap.vi*
- *Unflatten Pixmap.vi*
- *Draw Flattened Pixmap.vi*
- *Picture to Pixmap.vi*

Przykład 1 – Porównanie typów



Proste funkcje graficzne

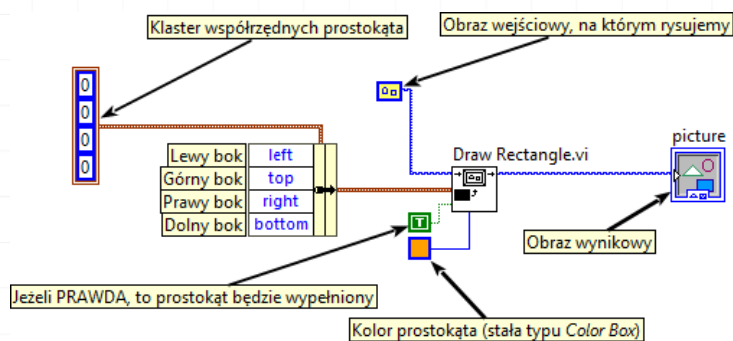


Korzystając z zestawu prostych funkcji, możliwe jest tworzenie grafik składających się z kropek, linii, prostych figur i tekstu. Niektóre funkcje wymagają podania wszystkich współrzędnych danego obiektu, inne rozpoczną rysowanie od bieżącego położenia kursora, które można zmieniać przy użyciu funkcji *Move Pen.vi*. Współrzędne zwykle przechowywane są w klastrach.

Poprzez odpowiednią opcję w *Property Node* możliwa jest programowa zmiana wymiarów wyświetlanej w indykatorze grafiki.

Korzystając z funkcji *Get Image Subset.vi* można przycinać grafikę do wymiarów podanego prostokąta, uzyskując wycinek dowolnego jej fragmentu.

Przykład 2 – Użycie funkcji *Draw Rectangle.vi*



Obsługa plików

Read JPEG File.vi



Read PNG File.vi



Read BMP File.vi



Write JPEG File.vi



Write PNG File.vi



Write BMP File.vi

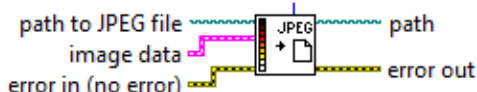


Read JPEG File.vi



Reads a JPEG file and creates the data necessary to display the file in a picture control.

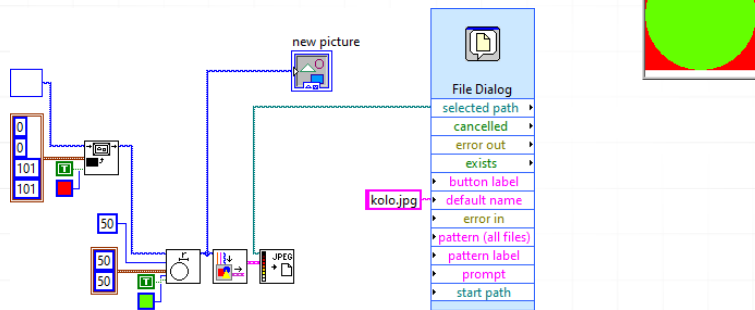
Write JPEG File.vi



Writes a JPEG file.

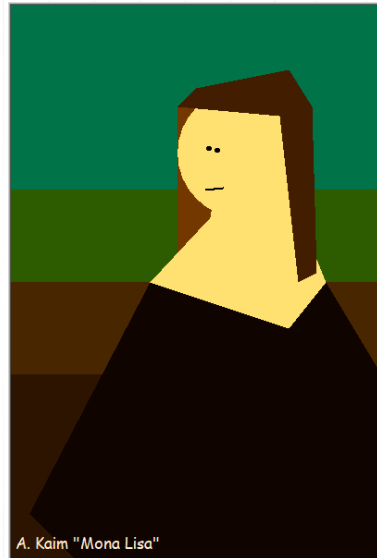
Podobnie jak w przypadku plików tekstowych, do obsługi grafik przygotowano zestaw prostych funkcji pozwalających na ich odczytywanie i zapisywanie w plikach formatu JPEG (JPG), PNG i BMP. Typ danych wykorzystywany przez te funkcje to *Image Data*.

Przykład 3 – Zapis do pliku



Zadanie 1

Stwórz program, który narysuje obrazek o rozdzielczości 600x400. W lewym dolnym rogu podpisz swoje dzieło. Wykorzystaj kilka różnych funkcji rysujących różne figury geometryczne.



Zadanie 2

Stwórz program, który wczyta Zajęcia 6 – Zadanie 2 (ściągnięty ze strony), a następnie zamieni jego kolory w następujący sposób: R->G, G->B, B->R.

Na Front Panelu wyświetl oba obrazki oraz zapisz zmodyfikowany obrazek na pulpicie pod nazwą zadanie2.jpg.

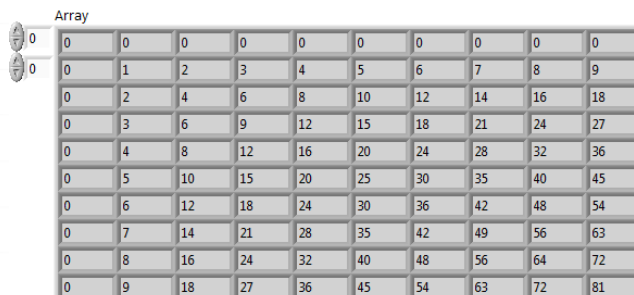
Wykorzystaj Auto-indexing oraz odpowiednie funkcje:



Zadanie 3

Stwórz tablicę dwuwymiarową 50x50, której kolejne elementy będą równe iloczynowi indeksów kolumny i wiersza. Następnie wyświetl taką tablicę jako obrazek na front panelu, przeskalowując go do rozdzielczości 500x500 pixeli.

Array



0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9
0	2	4	6	8	10	12	14	16	18
0	3	6	9	12	15	18	21	24	27
0	4	8	12	16	20	24	28	32	36
0	5	10	15	20	25	30	35	40	45
0	6	12	18	24	30	36	42	48	54
0	7	14	21	28	35	42	49	56	63
0	8	16	24	32	40	48	56	64	72
0	9	18	27	36	45	54	63	72	81

Zadanie 4

Pewien obraz poddany został scramblingowi przez przesunięcie każdej jego linii o pewną liczbę pikseli. Stwórz *descrambler*, który odczyta plik Zajecia 6 - Zadanie 4 IMG.bmp i przesunie każdą z linii o odpowiednią liczbę pikseli, zawartą w pliku Zajecia 6 - Zadanie 4 KEY.txt.

Wskazówka: wykorzystaj Auto-Indexing oraz funkcję „Rotate 1D Array”

Rotate 1D Array

