

Dariusz POJDA, Przemysław KOWALSKI  
Instytut Informatyki Teoretycznej i Stosowanej, PAN

## PROGRAMOWE PLATFORMY DLA ZAGADNIEŃ WIZJI KOMPUTEROWEJ<sup>1</sup>

**Streszczenie.** W artykule przedstawiono trzy środowiska, wykorzystywane dla prac w zakresie wizji komputerowej. Są to: Khoros, Matlab (z przybornikiem Image Processing) oraz Classman. Opisano i porównano ich możliwości, dostępność, zasady rozwoju, a także wydajność operacji wykonywanych z ich pomocą.

### A SOFTWARE PLATFORM'S FOR COMPUTER VISION PROBLEM

**Summary.** Article presents three software platforms used in computer vision. They are: Khoros, Matlab (with the Image Processing toolbox) and Classman. Their features, availability, development principles and efficiency were described and compared.

#### 1. Wstęp

Wraz z rozwojem wizji komputerowej pojawiło się wiele pakietów programowych, wspomagających przetwarzanie i analizę obrazów.

Najczęściej były one opracowywane przez ośrodki badawcze, realizujące badania w zakresie komputerowych systemów wizyjnych. Tak powstały Khoros, Classman czy Xite.

Inne pakiety rozwijały się w oparciu o programy, początkowo nie przeznaczone do przetwarzania obrazów, gdzie dodatkowe możliwości stanowiły odpowiedź na zapotrzebowanie rynku. Dobrym przykładem takiego pakietu jest Matlab.

Pojawiły się też pakiety specjalizowane – takie jak Midas, służący do przetwarzania obrazów w medycynie.

W artykule opisano trzy pakiety (Classman, Khoros, Matlab), używane oraz testowane przez Zespół Komputerowych Systemów Wizyjnych w IITiS PAN w Gliwicach.

Celem niniejszej pracy było wybranie platformy programowej dla realizowanych w zespole zadań, wynikających z grantu 8 T11C 049 12.

## 2. Wprowadzenie do programów

### 2.1. Pakiet Khoros

Khoros jest zorientowanym obiektowo środowiskiem służącym do graficznego tworzenia i uruchamiania algorytmów. Pakiet powstał w roku 1990 i jest nadal rozwijany. Początkowe prace nad projektem Khoros były prowadzone na Uniwersytecie Nowego Meksyku w Albuquerque. Rozwój pakietu spowodowany zapotrzebowaniem na tego rodzaju narzędzie stał się przyczyną założenia w roku 1993 konsorcjum, pod nazwą Khoral Inc, zajmującego się wyłącznie pracą nad pakietem, oraz jego dystrybucją. Siedzibą konsorcjum pozostaje nadal Uniwersytet Nowego Meksyku, natomiast jego udziałowcami są m.in. Departament Obrony USA, US Navy, Texas Instruments, Kodak i kilka innych mniej lub bardziej znanych firm.

Zasady dystrybucji pakietu, oraz łatwość jego obsługi sprzyjają jego akceptacji w środowiskach akademickich na świecie, gdzie jest używany jako narzędzie dydaktyczne, ułatwiające studentom zrozumienie metod przetwarzania obrazów. Pakiet rozprowadzany jest w dwóch wersjach: bezpłatnej, dostępnej w sieci internet przez FTP (użytkownik: anonymous) – w postaci źródłowej (Advanced Khoros), oraz komercyjnej, na płytach CD-ROM obejmującej zarówno źródła, jak i programy skompilowane dla kilku popularnych środowisk (Khoros Pro). Do wersji komercyjnej dołączona jest także pełna dokumentacja pakietu. Twórcy zakładają, że po zapoznaniu się z bezpłatną wersją pakietu klient dokona zakupu wersji komercyjnej, zyskując przy tym prócz dokumentacji także możliwość udziału w szkoleniach i kursach (płatnych) organizowanych przez Khoral Inc. Obecnie rozprowadzana jest wersja 2.2 pakietu, która ukazała się pod koniec 1997 roku. Wszystkie informacje praktyczne zawarte w tym opracowaniu dotyczą wcześniejszej wersji Advanced Khoros 2.1 (1996).

### 2.2. Pakiet Classman

Pakiet Classman został opracowany w Centre d'Automatique de Lille pod kierunkiem prof. Francois Cabestaing. Pierwsza wersja WinImage powstała w roku 1993, po

---

<sup>1</sup> Praca częściowo finansowana z grantu KBN 8 T11C 049 12.

przeniesieniu programu Image 1.2 do środowiska Windows. W roku 1994 wyodrębniono zarządzanie obiektami, przenosząc je do DLL'i, a sam program stał się ClassManagerem (Classman).

Celem było opracowanie programu, który pozwoliłby na testowanie algorytmów. Docelowe aplikacje powinny bazować na DLL'ach (bądź choćby ich kodzie), nie wykorzystując jednak samego programu zarządzającego, i jego interfejsu użytkownika.

Opracowano wiele dodatkowych funkcji: do analizy obrazów stereo, obrazów kolorowych, etc.

Przewiduje się przeniesienie programu do wersji 32-bitowej Windows, a opracowywane zgodnie z zaleceniami funkcje powinny być łatwo przenaszalne do systemu UNIX.

### 2.3. Matlab

Program Matlab został opracowany przez Clewa Molera w roku 1980, początkowo w języku Fortran. Nazwa Matlab jest skrótem od **M**atrix **L**aboratory (laboratorium macierzowe) – podstawowym typem danych, przetwarzanych przez Matlaba są macierze. Kolejna wersja programu została opracowana w firmie The MathWorks Inc. w języku C, aczkolwiek do dziś można znaleźć w Matlabie pewne rozwiązania wywodzące się z cech języka Fortran. Matlab był sukcesywnie rozwijany – powstawały nowe wersje, pojawiło się również wiele przydatnych przyborników (ang. *Toolboxes*). Najnowsza wersja programu nosi numer 5.2 i wykorzystuje przybornik Image Processing 2.1; w artykule opisano jednak wersję wcześniejszą 4.2c.1, która była testowana przez Zespół Komputerowych Systemów Wizyjnych, wyposażoną w przybornik Image Processing 1.0b, wspomagający Matlaba przy zadaniach przetwarzania obrazu.

Pakiet Matlab pracuje w wielu różnych środowiskach, od domowego PC, po superkomputery, co sprzyja jego popularyzacji.

Pojawiło się wiele publikacji dotyczących Matlaba ([1], [2], [3]), a także darmowo rozprowadzono „klon” zwany Octave.

Tabela I

#### Dystrybucja

| Pakiet             | Khoros                        | Classman                  | Matlab   |
|--------------------|-------------------------------|---------------------------|--|
| Producent          | Khoral Inc.                   | CAL                       | The MathWorks  |
| Cena (dystrybucja) | 549\$ lub darmowo (przez ftp) | Do uzgodnienia z autorami | Zależy od konfiguracji, orientacyjnie dla systemu UNIX: 830\$, przyborniki po 345\$. |

### 3. Założenia platformy

Powstawaniu programów towarzyszyły różne założenia i oczekiwania. Inaczej wyglądał rozwój narzędzia zaprojektowanego dla operacji macierzowych Matlab, inaczej projektowanych od początku dla przetwarzania obrazów Khorosa i Classmana.

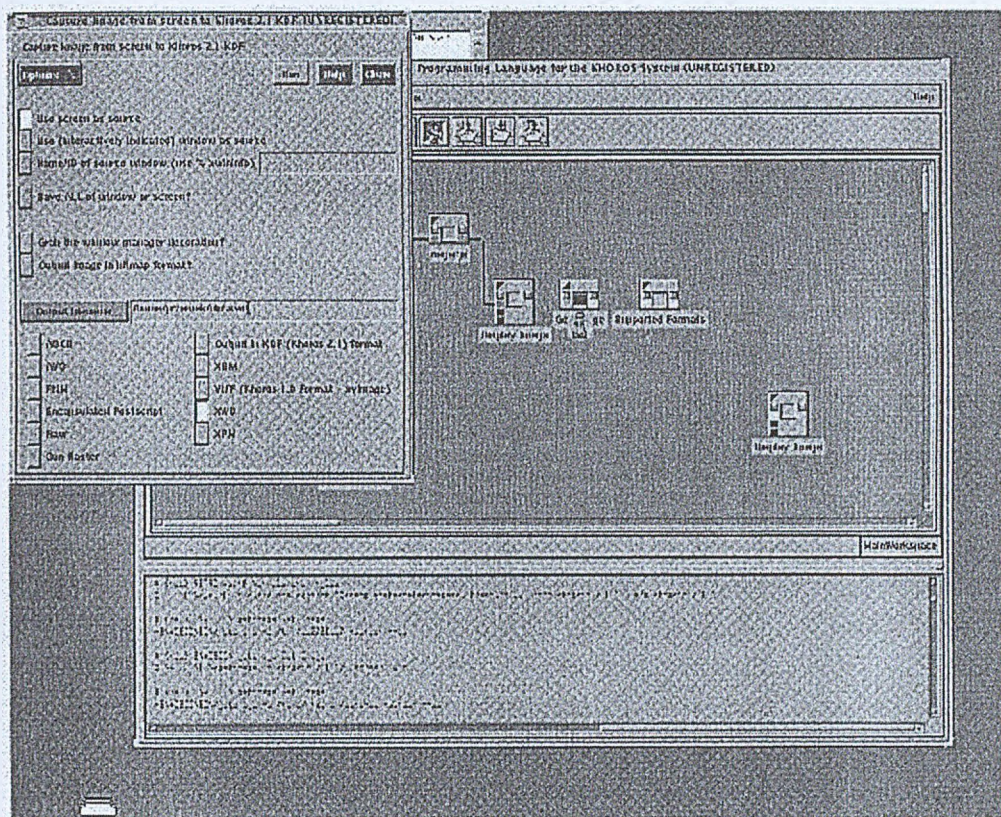
#### 3.1. Khoros

Pakiet Khoros składa się z wielu niezależnych od siebie programów. Najistotniejsze znaczenie dla programisty pracującego z nim mają trzy z nich:

**Cantata** – graficzny interfejs ułatwiający tworzenie i uruchamianie programów w Khorosie. Programując w środowisku oferowanym przez program cantata tworzy się tak zwane przestrzenie robocze (ang. *workspaces*), które są graficzną reprezentacją skryptów interpretowanych przez cantatę. Każda procedura wywoływana przez cantatę jest w przestrzeni roboczej reprezentowana przez odpowiadającą jej ikonę (ang. *glyph*). Ikony połączone są ścieżkami przepływu danych. Główną cechą odróżniającą Khorosa od innych (tradycyjnych) platform programowych jest przebieg wykonywania programów sterowany przepływem danych. To znaczy każda operacja jest wykonywana wtedy, kiedy dostępne są wszystkie dane wejściowe niezbędne dla jej wykonania. Procesy są wykonywane współbieżnie, o ile nie jest konieczna ich wzajemna synchronizacja. Istnieje też możliwość rozproszenia obliczeń na kilku pracujących równolegle maszynach.

**Craftsman** – umożliwia dodawanie do pakietu nowych przyborników (ang. *toolboxes*), a także zarządzanie zawartością istniejących.

**Composer** – narzędzie to służy do zarządzania projektami podczas tworzenia nowych lub edycji istniejących procedur. Composer ułatwia kontrolę nad poszczególnymi elementami projektu, takimi jak kod źródłowy procedury, jej ikona i związane z nią okno dialogowe, służące do określania parametrów wywołania procedury z poziomu cantaty, czy też dokumentacja.



Rys. 1. Ekran pracy komputera, z programem Cantata – widać otwarte okno parametrów polecenia

Fig. 1. Computer screen, with running Cantata – parameters window open

Jak wspomniano, procedury dostępne programiście są pogrupowane w przybornikach. Standardowo z pakietem rozprowadzane są między innymi przyborniki:

- Arithmetic – udostępniający podstawowe operacje arytmetyczne,
- Image Proc – procedury przetwarzania obrazów,
- Input Output – zarządzanie odczytem i zapisem różnych formatów danych,
- Matrix – operacje na macierzach,
- Visualisation – wizualizacja danych

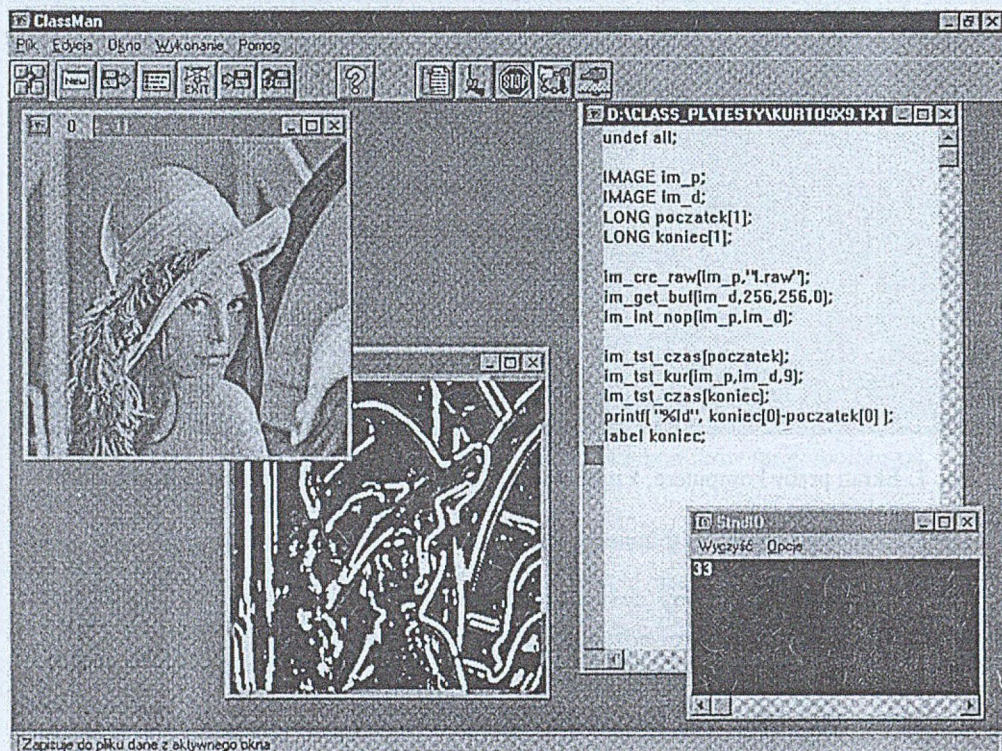
Dla użytkowników komercyjnej wersji Khorosa dostępne są również inne przyborniki, w tym także umożliwiające przenoszenie procedur pomiędzy Khorosem i Matlabem. Wiele innych przyborników można również znaleźć w sieci internet, gdzie udostępniają je różnego rodzaju ośrodki badawcze wykorzystujące Khorosa w swoich pracach.

Procedury Khorosa mają postać plików wykonywalnych systemu UNIX, mogą więc to być również dobrze skompilowane programy napisane w języku C lub Fortran, jak również skrypty

wykonywane przez powłokę systemu. Pakiet wspomaga tworzenie własnych procedur w języku C. Ze względu na możliwość wykonywania procedur (programów) Khorosa bezpośrednio z linii poleceń można je wykorzystywać także w skryptach tworzonych poza pakietem.

### 3.2. Classman

Classman jest aplikacją MDI pracującą w środowisku Windows. Możemy ją łatwo rozbudować, tworząc biblioteki DLL, zawierające potrzebne funkcje. Funkcje te są następnie wywoływane z poziomu makra.



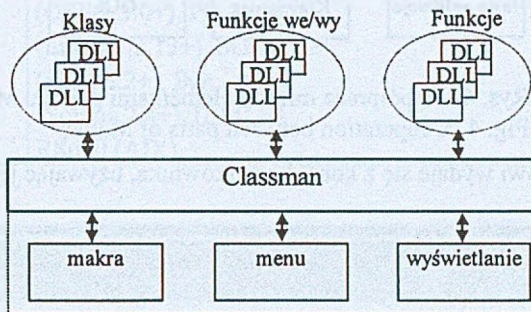
Rys. 2. Ekran komputera podczas pracy programu Classman, wyświetlany jest obraz „Lena” oraz ten sam obraz, po filtracji nieliniowej kurtozą, w dwóch oknach do prezentacji grafiki

Fig. 2. Computer screen with running Classman; image „Lena”, and the same image filtered by nonlinear filter (with kurtosis) are shown in two graphics windows

Istnieją trzy rodzaje bibliotek DLL współpracujących z Classmanem:

- biblioteki opisujące klasy (buffer, masque, histogramme, memoire),
- biblioteki zawierające funkcje wejścia/wyjścia dla klas,
- biblioteki zawierające funkcje.

Program powstał w języku C – także funkcje powstają w tym języku.



Rys. 3. Założenia pakietu Classman

Fig. 3. Essential structure of Classman

### 3.3. Matlab

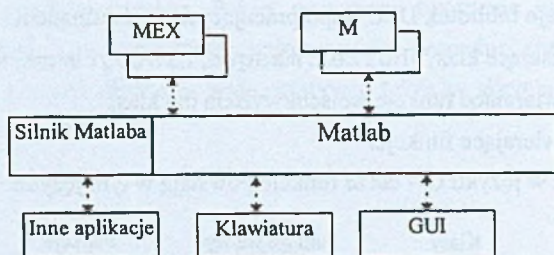
W przypadku Matlabu mamy do czynienia ze środowiskiem, którego elementami są program Matlab oraz przyborniki. Przyborniki stanowią zbiory funkcji rozszerzających możliwości programu. Obecnie pozwalają one m.in. na:

- modelowanie układów,
- wspomaganie pracy przy obliczeniach statystycznych,
- przetwarzanie sygnałów,
- przetwarzanie obrazów,
- pracę z sieciami neuronowymi,
- wykonywanie obliczeń symbolicznych.

Dodatkowe funkcje dla Matlabu można zaimplementować jako:

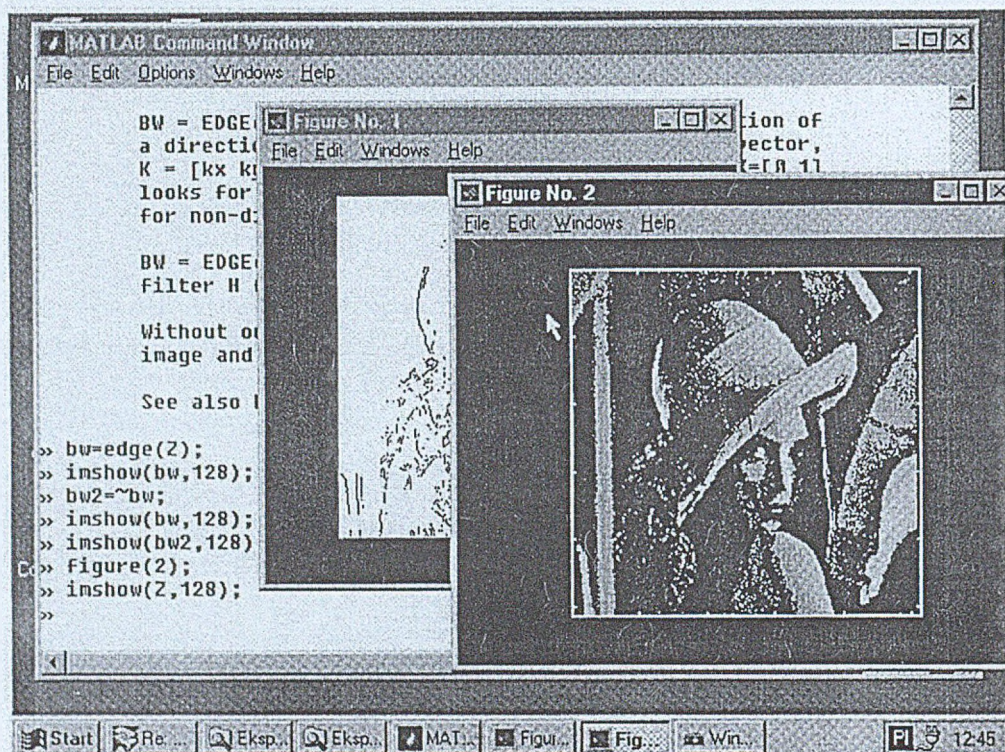
- m-pliki (czyli funkcje w języku makr Matlabu – opisane w [2] i [3]),
- pliki mex, czyli pliki binarne, przygotowane w C lub Fortranie.

Do poleceń Matlabu można także odwoływać się z poziomu innych aplikacji, jako odwołania do „silnika Matlabu” (ang. Matlab Engine).



Rys. 4. Współpraca między elementami pakietu Matlab  
 Fig. 4. Cooperation between parts of Matlab

Polecenia Matlabowi wydaje się z konsoli użytkownika, używając języka jego skryptów.



Rys. 5. Ekran komputera podczas pracy programu Matlab, wyświetlany jest obraz „Lena” oraz wykryte (na tym obrazie) krawędzie, w dwóch oknach do prezentacji grafiki  
 Fig. 5. Computer screen with running Matlab; image „Lena”, and detected edges (from the same image) are shown in two graphics windows



Tabela 2

## Dostępność

| Pakiet     | Khoros  | Classman       | Matlab  |
|------------|---|----------------|---|
| Środowisko | UNIX wraz z X11R5+, czyli na maszynach: DEC (OSF/1 3.0+), PC (Linux 1.2.13+), SGI (IRIX 5.2+), Sun (Solaris 2.4+), IBM RS600 (AIX). | Windows (3.1+) | Sun4, Solaris, HP-UX, IRIX, IRIX64, AIX, Digital UNIX, Linux, MacOS, Windows, DOS (stare wersje). |

#### 4. Reprezentacja obrazów

Dla każdego pakietu można wskazać wewnętrzne sposoby reprezentacji obrazów, jak i rodzaje plików, z którymi współpracuje.

Sposoby reprezentacji wskazują na sposób wykorzystania informacji obrazowej.

Typy plików natomiast określają możliwości wymiany danych z innymi programami.

Tabela 3

## Rodzaje plików graficznych

| Pakiet                      | Khoros   | Classman  | Matlab  |
|-----------------------------|--|---|---|
| Rodzaje plików              | kdf, viff (własne Khorosa), raw, avs, arf, pnm, SunRaster, xbm, xwd, xpm | bmp, jpg, tiff, raw, rgb, ima                                   | gif, tiff, bmp, hdf, pcx, xwd <sup>2</sup>                  |
| Sposób reprezentacji obrazu | Kobiekt (obiekty polimorficzne)  | Obrazy w stopniach szarości, kolorowy indeksowany lub kolorowy. | Obraz w stopniach szarości, obraz indeksowany, macierze RGB |

<sup>2</sup> Niektóre rodzaje plików mają swoje warianty, a tylko część tych wariantów jest obsługiwana przez Matlaba – dobrym przykładem jest typ BMP, tylko warianty opisujące piksel czterema i ośmioma bitami są rozpoznawane przez Matlaba.

#### 4.1. Khoros

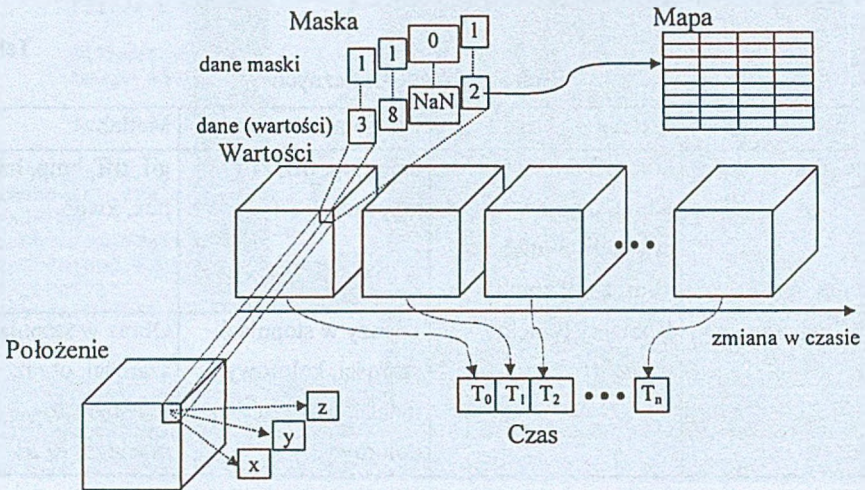
Obraz jest w Khorosie reprezentowany przez kobjekty. Kobjekty są złożonymi obiektami, pozwalającymi na polimorficzną reprezentację obrazu.

Kobjekt składa się z pięciu segmentów.

Tabela 4

Segmety kobjektu

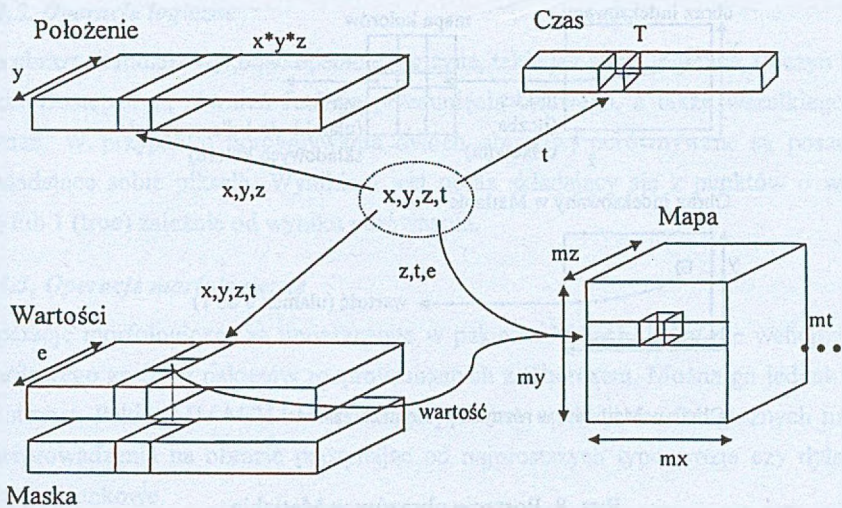
| Nazwa                                  | Wymiary | Zawartość                   | Uwagi  |
|--|---------|-----------------------------|--|
| Wartości (ang. <i>Value</i> )          | $5W^3$  | Wartości pikseli            | Typy danych – m.in.: byte, int, double, complex      |
| Maska (ang. <i>Mask</i> <sup>3</sup> ) | $5W$    | Poprawność wartości piksela | 0 – nieprawidłowe (Inf, Nan),<br>1 – dane prawidłowe |
| Położenie (ang. <i>Location</i> )      | $3W$    | Położenie (x,y,z) piksela   | Ważne, gdy próbkowanie było nierównomierne           |
| Mapa (ang. <i>Map</i> )                | $5W$    | Mapa kolorów                |  |
| Czas (ang. <i>Time</i> )               | $1W$    | Znacznik czasu dla próbki   | Ważne, gdy próbkowano przy zmiennym kwancie czasu    |



Rys. 6. Segmety składające się na kobjekt

Fig. 6. Segments of a kobjekt

<sup>3</sup> Trzy wymiary przestrzenne (x,y,z), czas (t) oraz element wektora (e).



Rys. 7. Wzajemne relacje pomiędzy danymi z segmentów obiektu  
 Fig. 7. Mutual relations between data from object's segments

#### 4.2. Classman

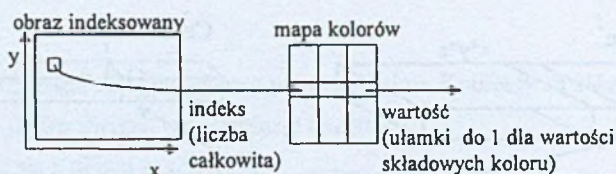
Obraz w Classmanie może być zapisany na jeden z pięciu sposobów:

- obraz w odcieniach szarości (1 piksel opisano przez 1 bajt),
- obraz kolorowy, indeksowany (1 piksel opisany został 1-bajtowym indeksem),
- obraz kolorowy (tu trzy warianty):
- 1 piksel opisany trzema bajtami,
- 1 piksel opisany dwoma bajtami oraz
- 1 piksel opisany przez cztery bajty.

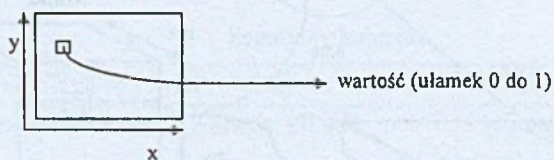
#### 4.3. Matlab

Postacie obrazów w pamięci dla Matlab'a są następujące:

- obraz indeksowy – każdy piksel zawiera odwołanie do mapy kolorów (standardowa postać dla odczytu danych z pliku),
- obraz o różnych poziomach szarości – macierz, której elementy odpowiadają pikselom – wartość 1 odpowiada kolorowi białemu, 0 – czarnemu, wartości pośrednie reprezentują różne stopnie szarości,
- obraz czarno-biały – j.w., ale dopuszczalne są tylko dwie wartości – 0 i 1,
- macierze RGB lub HSV.



Obraz indeksowany w Matlabie



Obraz w Matlabie, w różnych poziomach szarości

Rys. 8. Postacie obrazów w Matlabie

Fig. 8. Forms of images in Matlab

Praktycznie jednak większość operacji związanych z przetwarzaniem obrazów odbywa się wyłącznie na macierzach intensywności. Oczywiście istnieją możliwości konwersji pomiędzy tymi sposobami reprezentacji obrazu.

## 5. Możliwości programów

### 5.1. Khoros

#### 5.1.1. Operacje na paletach kolorów

Ponieważ obrazy w Khorosie są traktowane dokładnie tak samo jak dowolne inne dane, możliwe jest wykonywanie na nich dowolnych operacji, w tym arytmetycznych. Można więc obrazy dodawać, mnożyć, logarytmować. Należy tu jednak pamiętać o tym, że wyniki takich operacji mogą być różne, zależnie od typu obrazu (pojedynczy piksel może być w Khorosie reprezentowany na wiele sposobów – od bajtu bez znaku, po wartości typu double). Wszelkie operacje arytmetyczne mogą być przeprowadzane skalarnie (pakiet Arithmetic), jak również macierzowo (pakiet Matrix).

### 5.1.2. Operacje logiczne

Na obrazach można wykonać operacje logiczne, takie jak suma logiczna, iloczyn logiczny, czy xor. Dostępne są również funkcje przesunięcia bitowego, a także wszelkiego rodzaju porównań. W przypadku porównywania dwóch obrazów, porównywane są poszczególne, odpowiadające sobie piksele. Wynikiem jest obraz składający się z punktów o wartości 0 (false) lub 1 (true) zależnie od wyniku porównania.

### 5.1.3. Operacje morfologiczne

Operacje morfologiczne są umieszczone w pakiecie Mmach, który nie wchodzi w skład standardowego zestawu pakietów rozprowadzanych z Khorosem. Można go jednak pobrać z sieci internet. Pakiet MMACH zawiera bogaty zestaw operacji morfologicznych możliwych do przeprowadzenia na obrazie poczynając od najprostszych typu erozja czy dylatacja, po operacje warunkowe.

### 5.1.4. Statystyka 2-D

Funkcje statystyczne zostały umieszczone w jednej procedurze (Statistics). Wyboru właściwej funkcji dokonuje się przez ustawienie odpowiedniego argumentu wejściowego. Dostępne są m.in. mediana, wariancja, odchylenie standardowe, kurtoza, maksimum i minimum. Funkcje statystyczne są obliczane dla całego obrazu lub dla jego części, którą wyznacza maska operacji.

### 5.1.5. Praca rozproszona, sterowanie przepływem danych

Khoros wykorzystuje wiele mechanizmów oferowanych przez system UNIX. Procesy, o ile to możliwe, wykonywane są współbieżnie. Przebieg algorytmu jest sterowany przepływem danych, to znaczy dana operacja wykonywana jest (o ile projektant nie zdecyduje inaczej) wtedy, kiedy dostępne są wszystkie dane wejściowe niezbędne dla jej wykonania. Istnieje także możliwość wykorzystania większej liczby komputerów pracujących równolegle w sieci, w takim wypadku rola projektanta algorytmu ogranicza się do zdefiniowania zdalnych maszyn dostępnych w sieci, oraz uruchomienia na każdej z nich procesu phantomd. Całość operacji związanych z rozdziałem procesów pomiędzy poszczególne maszyny wykonywana jest przez system niezależnie od użytkownika. W pracy lokalnej dane pomiędzy procesami są standardowo przesyłane przy użyciu plików tymczasowych. Możliwe także jest wykorzystanie pamięci dzielonej (ang. *shared memory*), potoków (ang. *streams*), oraz w przypadku pracy rozproszonej gniazd (ang. *sockets*).

## 5.2. Classman

Operacje Classmana zdefiniowano w kilku bibliotekach klasowych. Są to:

- `buffer.dll` – umożliwia odczyt, zapis, wyświetlanie, oraz modyfikację (rysowanie prostych obiektów geometrycznych) na obrazie;
- `masque.dll` – tworzenie maski; istnieją cztery rodzaje masek: splotowa, korelacyjna, morfologiczna binarna, morfologiczna o stopniach szarości;
- `histo.dll` – zawiera opis struktury histogramu.

Poza bibliotekami klasowymi utworzono również wiele bibliotek funkcyjnych.

### 5.2.1. Operacje na palecie kolorów

Obrazy w Classmanie można przeskalować, dodając do każdej wartości bądź odejmując pewną liczbę, lub mnożąc przez zadany składnik. Obrazy można też przez siebie przemnożyć (ze znakiem lub bez) lub podzielić.

### 5.2.2. Operacje filtrowania

Wśród masek mamy też maski splotowe, które składają się z elementów całkowitych (-255 do 255) oraz liczby normującej. Operację splotu można wykonać ze znakiem i bez znaku.

Pewne filtry predefiniowano:

- uśredniający (3x3),
- laplasjanowy (3x3, z uwzględnieniem 4 lub 8 sąsiadów),
- gradientowy (3x3, poziomy lub pionowy, typu 1 lub 2),
- Prewitta (3x3),
- Sobela (3x3, filtracja dwumaskowa),
- medianowy (także jego warianty dla sąsiadów w poziomie i w pionie).

### 5.2.3. Operacje na histogramach

Można obliczyć histogram dla danego obrazu. Istnieją różne typy histogramów, odpowiadające różnym typom obrazów (operujące na poziomach szarości, kolorach indeksowanych etc.). Można wykonać operację wyrównania histogramu.

### 5.2.4. Segmentacja i krawędzie

Istnieje operacja binaryzacji. W oparciu o obraz binarny można dokonać analizy połączeń, zaznaczając regiony (maksymalnie 1024 regiony).

W Classmanie zaimplementowano operacje znajdowania krawędzi za pomocą filtrów splotowych (Prewitta, Sobela), oraz zdefiniowano filtrację Dericha.

Kontury zapisuje się w specjalnych strukturach i odpowiadającym im plikach \*.edg. Można utworzyć wielokąty konturu metodą Pavlidisa.

### 5.2.5. Operacje morfologiczne

Zdefiniowano operacje dylatacji i erozji, oraz pogrubiania i odchudzania. Operacje morfologiczne można również samodzielnie definiować – służą do tego maski morfologiczne (dla obrazów czarno-białych bądź też o różnych poziomach szarości).

### 5.2.6. Statystyka 2-D

Dla każdego obrazu można uzyskać wartości minimum i maksimum.

### 5.2.7. Przekształcenia obrazów

Możliwe są różne transformacje obrazu na podstawie przesunięć opisanych w drugim obrazie.

## 5.3. Matlab

Możliwości Matlabu są bardzo różnorodne. Różnorodność ta ma swe źródło w dużej liczbie oferowanych przybowników, umożliwiających wykorzystanie Matlabu w wielu różnych dziedzinach. Do przetwarzania obrazów przygotowano przybownik Image Processing.

### 5.3.1. Operacja na kolorach

Wśród operacji zdefiniowanych w przybowniku Image Processing znajdują się oczywiście polecenia operujące na kolorach (stopniach szarości i nie tylko) obrazu. Do operacji tych zaliczyć można: przeprowadzenie korekcji gamma, rozjaśnienie/przyciemnienie obrazu, zwiększenie jego intensywności, zmianę liczby kolorów.

Obraz (tak jak każdą macierz w Matlabie) można przeskalować (dodając, odejmując, mnożąc lub dzieląc przez liczbę).

### 5.3.2. Operacje logiczne

Obrazy w Matlabie są reprezentowane przez macierze. I tak jak na innych macierzach, można wykonywać na nich operacje sumy logicznej, iloczynu logicznego, xor-a, czy negacji.

### 5.3.3. Filtry

Można dokonać filtracji, wykorzystując dowolny zaprojektowany przez nas filtr splotowy – ograniczeniem jest brak dzielnika (istnieją jednak proste operacje dzielenia i mnożenia macierzy przez skalar, które mogą zastąpić dzielnik). Kilka filtrów predefiniowano (filtr gaussowski, Sobela, Prewitta, Laplasa, laplasjan z filtru gaussowskiego, filtr uśredniający i filtr podkreślający kontrastowość).

Filtrowanie splotowe można ograniczyć do obszaru wskazanego przez odpowiednio utworzoną maskę.

Możemy również wykonywać operacje filtrowania nieliniowego dla dowolnej, wymyślonej przez użytkownika funkcji – operacje te są jednak zwykle czasochłonne.

Wśród innych filtrów nieliniowych predefiniowano filtr medianowy oraz dolnoprzepustowy, adaptacyjny filtr Wienera.

Korzystając z pakietu Matlab, możemy dokonać analizy odpowiedzi częstotliwościowej zadanego filtru; co więcej – możemy zaprojektować filtr dla zadanego pasma odpowiedzi częstotliwościowej. Operacje związane z odpowiedzią częstotliwościową i tworzeniem filtrów w oparciu o dane okno częstotliwości związane są z operacjami zdefiniowanymi w pakiecie Signal Processing.

#### **5.3.4. Operacje na histogramach**

Lista operacji na histogramach w przypadku Matlab jest krótka. Można jedynie wyświetlić histogram, oraz przekształcić obraz do postaci określonej przez zadany histogram. Ta z pozoru prosta funkcja pozwala na stosunkowo łatwe wykonywanie bardzo wielu operacji histogramowych. Brak określenia docelowego histogramu powoduje wykonanie domyślnej operacji – wyrównania histogramu.

#### **5.3.5. Wsparcie dla segmentacji i wykrywanie krawędzi**

Matlab zasadniczo nie wspiera segmentacji – zdefiniowano jedynie jedną z operacji konwersji z macierzy intensywności do obrazu czarno-białego jako operację progowania przy zadanym progu.

W przypadku przyborka Image Processing zdefiniowano cztery metody wykrywania krawędzi: Sobela, Prewitta, Robertsa oraz Marra-Hildretha.

#### **5.3.6. Operacje morfologiczne**

Pakiet Matlab zawiera kilka funkcji morfologicznych. Są to operacje erozji i dylatacji (mające kilka wariantów), operacja perimeter (czyli ekstrakcja krawędzi), oraz funkcja „operacja morfologiczna” (`bwmorph`) z wieloma predefiniowanymi operacjami.

Wszystkie operacje morfologiczne w Matlabie wykonywane są na obrazach czarno-białych; a każda z operacji zdefiniowana jest dla otoczenia  $3 \times 3$ . Definiowanie własnych operacji morfologicznych jest dość uciążliwe – należy przygotować oczekiwaną wartość piksela dla wszystkich 512 możliwych stanów otoczenia  $3 \times 3$ .



### 5.3.7. Statystyka

W Matlabie istnieje wiele funkcji statystycznych – zwłaszcza w przyborniku Statistical Toolbox. Lecz tylko nieliczne funkcje są przystosowane do pracy dla macierzy dwuwymiarowych – większość operuje na wektorach.

Dla macierzy 2-D zaimplementowano średnią, standardowe odchylenie oraz korelacje dwu macierzy. Określenie maksimum oraz minimum również jest łatwe – by uzyskać dwuwymiarowe maksimum, wystarczy napisać –  $\max(\max(\text{obraz}))$ .

### 5.3.8. Transformaty

Zwykła transformata Fouriera została dla Matlabu opracowana jedynie dla danych jednowymiarowych. Listę transformat zamieszczono w tabelce.

### 5.3.9. Obszary zainteresowań i przekształcenia

Korzystając z przybornika Image Processing możemy definiować obszary zainteresowania – zarówno określając pewną powierzchnię, jak i wybierając kolor.

Matlab pozwala też na przeskalowanie obrazu i jego obrót.

Tabela 5

Dostępne transformaty

| Pakiet                | Khoros        | Classman | Matlab                                   |
|-----------------------|---------------|----------|--|
| Dostępne transformaty | FFT, Hadamard | 4        | FFT, FFT2, Radon, DCT, IDCT, IFFT2, IFFT |

## 6. Wizualizacja

Ostatnim etapem przetwarzania obrazów jest prezentacja wyników. Wyniki powinny być czytelne dla użytkownika.

### 6.1. Khoros

W pakiecie Khoros można znaleźć wiele funkcji odpowiedzialnych za wizualizację – jest to cała kategoria wizualizacja (ang. visualisation) dostępna z programu cantata. Możemy wyświetlać obrazy, wykresy lub animacje.

<sup>4</sup> Opracowano transformaty (np. FFT) dla realizacji niektórych projektów.

## 6.2. Classman

Można wyświetlić obrazy we wszystkich dostępnych reprezentacjach. Dodatkową możliwością jest dodawanie elementów grafiki wektorowej do wyświetlanego obrazu.

## 6.3. Matlab

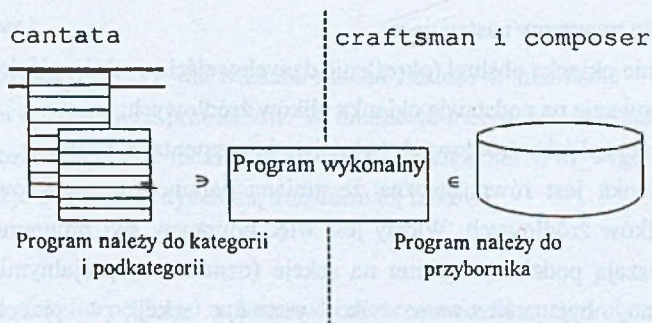
Matlab potrafi wyświetlić obraz w każdej z wykorzystywanej przez niego postaci. Sposoby wyświetlania obrazów różnią się od typowych sposobów wyświetlania macierzy (obraz jest inaczej ułożony). Można wyświetlać również kontur (czyli krawędzie obszarów o określonym kolorze). Wyświetlanie można połączyć z przeskalowaniem do pełnej palety kolorów. Wykorzystując fakt, że obraz jest macierzą, można zastosować także typowe funkcje służące do wyświetlania macierzy (należy jednak pamiętać o różnicach w ułożeniu obrazu oraz interpretacji elementów macierzy).

# 7. Rozwój oprogramowania w oparciu o pakiety

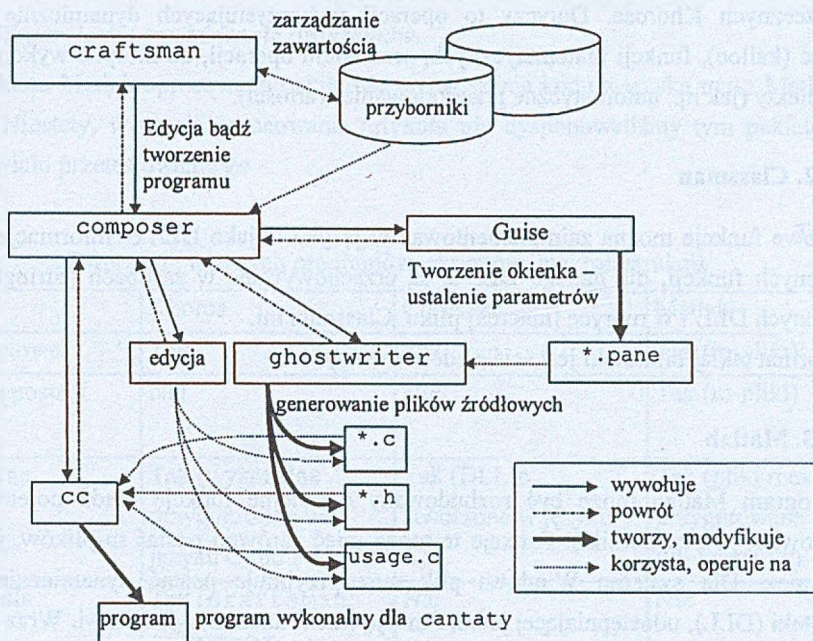
Lista potrzebnych operacji nigdy nie może być uznana za pełną, zwłaszcza w szybko rozwijających się dziedzinach, takich jak przetwarzanie obrazów. Stąd każdy pakiet powinien mieć możliwości dalszego rozwoju.

## 7.1. Khoros

Dodatkowe operacje dla pakietu Khoros tworzone są jako skrypty lub w języku C. Istnieje oprogramowanie wspomagające tworzenie nowych operacji. Operacje zebrane są w przybornikach. Oprócz podziału na przyborniki, istnieje podział na kategorie. Podział na kategorie wskazuje na położenie operacji w menu aplikacji cantata.



Rys. 9. Każdy program należy do kategorii, podkategorii i przybornika  
 Fig. 9. Every program belongs to a category, subcategory and toolbox



Rys. 10. Tworzenie nowych programów pakietu Khoros  
 Fig. 10. Creating new programs for Khoros

Każda funkcja jest w rzeczywistości niezależnym programem wykonywalnym i może być wykorzystywana także poza pakietem. Rozróżniane są dwa rodzaje funkcji/programów:

- xvroutines, które zawsze korzystają z interfejsu graficznego (wymagają X-Windows),
- vroutines, które nie wykorzystują interfejsu graficznego, o ile są wywołane z linii poleceń.

Nową funkcję tworzymy następująco:

- utworzenie okienka obsługi (określenie danych wejściowych i wyjściowych operacji);
- wygenerowanie na podstawie okienka plików źródłowych;
- modyfikacja plików źródłowych (właściwa implementacja operacji).

Zmiana okienka jest równoznaczna ze zmianą parametrów wejściowych i wymusza modyfikację plików źródłowych. Ważny jest więc poprawny styl programowania – twórcy Khorosa wymuszają podział programu na sekcje (oznaczone specjalnymi znacznikami) – operacje powinny być realizowane tylko wewnątrz sekcji; w przeciwnym wypadku modyfikacje wprowadzone przez ghostwritera<sup>5</sup> uszkodzą kod programu.

W praktyce, wiele operacji wykonywanych jest przez odwołanie się do funkcji bibliotecznych Khorosa. Dotyczy to operacji wykorzystujących dynamicznie alokowaną pamięć (kalloc), funkcji matematycznych, oraz wielu operacji, do których wykonania zdolne są kobiekty (jak np. automatyczne przeskalowanie wartości).

## 7.2. Classman

Nowe funkcje można zaimplementować w języku C jako DLL'e. Informacje o zestawie dostępnych funkcji, dla danego DLL'a, są przechowywane w zasobach (Stringtable). Lista dostępnych DLL'i w rubryce [macros] pliku Classman.ini.

Format takiej biblioteki jest ściśle zdefiniowany.

## 7.3. Matlab

Program Matlab może być rozbudowany o kolejne funkcję (bądź pojedyncze, bądź zgrupowane w przyborniki). Funkcje te mogą mieć zarówno postać m-plików, jak i plików typu mex. Dla systemu Windows plik mex przyjmuje postać dynamicznie ładowanej biblioteki (DLL), udostępniającej jedną funkcję (mexFunction) Matlabowi. Wraz z Matlabem typowa instalacja zawiera zestaw plików wykorzystywanych przy kompilacji plików mex w językach Fortran i C.

Istnieje generator kodu C, który tworzy plik w C na podstawie makra Matlaba.

W testowanej wersji pojawiły się problemy z przetwarzaniem w C dużych macierzy (a takie dominują przy pracy z obrazami). Być może źródło problemów leżało po stronie kompilatora C, nie mniej było to duże utrudnienie dla pracy z Matlabem, w środowisku Windows.

---

<sup>5</sup> Dla xvoutines i specyfikacji okienka \*.form, właściwym generatorem programów jest nie ghostwriter, ale conductor. Zachowuje się on jednak inaczej, w czasie generowania bądź modyfikowania programów.

### 7.3.1. Internet

Wiele dodatkowych funkcji dla Matlaba można znaleźć w internecie – często stanowią one ilustrację dla opisywanych problemów. W internecie można znaleźć także wiele pełnych przybory rozszerzających możliwości Matlaba, takich jak Uwi\_Vigo – rozszerzający Matlab o operacje związane z dyskretną transformatą falkową.

### 7.3.2. GUI

Matlab może również posłużyć za kanwę do opracowania własnego roboczego stanowiska – umożliwi on utworzenie własnego menu, przycisków itd. Umożliwi też odczyt wskazań myszki, odczyt wartości piksela wskazanego przez myszkę, oraz zaznaczenie myszką obszaru.

### 7.3.3. Wsparcie dla generowania programów

Dla pakietu Matlab opracowano pakiet dla przenoszenia kodu z języka makr Matlab do języka C. Niestety, w czasie opracowania artykułu nie dysponowaliśmy tym pakietem, co uniemożliwiło przetestowanie go.

Tabela 6

Tworzenie własnych programów, skryptów, makroinstrukcji

|                                       | Khoros  | Classman                         | Matlab  |
|---------------------------------------|---|----------------------------------|---|
| Pliki skryptowe                       | Tak   | Tak                              | Tak (m-pliki)                                   |
| Funkcje w postaci skryptu             | Nie   | Nie                              | Tak (m-pliki)                                   |
| Pliki binarne                         | Tak (wykonalne przygotowane w języku C lub Fortran)             | Tak (DLL'e stworzone w języku C) | Tak (pliki mex, przygotowane w C lub Fortranie) |
| Wsparcie dla tworzenia oprogramowania | Tak (craftsman, composer, ghostwriter, conductor, dla języka C) | Nie                              | Nie   |
| Wsparcie dla interfejsu użytkownika   | Tylko okno dla „klocka”   | Jak zwykły program w C           | Tak (dla skryptów), lub jak zwykły program w C  |

## 8. Uwagi, porady

Podczas pracy z pakietem użytkownik uczy się unikania najczęstszych błędów, poznaje sposoby pozwalające wydajniej pracować. Pozwala to zebrać porady praktyczne dla innych użytkowników.

### 8.1. Khoros

Jeśli za istotny parametr uznajemy zajętość pamięci – warto zadeklarować odwołania do bibliotek Khorosa, jako odwołania dynamiczne. Zmniejsza to zajętość pamięci dla pojedynczego programu o około 1.6MB (dla komputera Sun).

Podczas pracy zdalnej (np. za pośrednictwem programu Exceed) należy zwrócić uwagę, że standardowa operacja wyświetlania działa prawidłowo jedynie dla ekranu pracującego w trybie 256-kolorowym.

### 8.2. Matlab

Matlab przystosowany jest do pracy interakcyjnej – wykonanie pojedynczej operacji kończy się wyświetleniem jej wyniku. W przypadku operacji na obrazach oznacza to wyświetlenie macierzy liczb (o rozmiarach obrazu) – by tego uniknąć należy blokować wyświetlanie wyniku (pisząc średnik po poleceniu).

W celu podniesienia wydajności wykonywania m-plików zaleca się, aby nie wykorzystywać pętli, tam gdzie można wykonać operację na macierzy, oraz w miarę możliwości dokonywać prealokacji zmiennych.

W testowanej wersji Matlaba pojawiły się problemy z przetwarzaniem obrazów RGB – istniały trudności z ich konwersją i wyświetlaniem.

## 9. Wydajności programów

W celu przeprowadzenia porównania wydajności programów wykonano serię następujących testów:

- a) szybka transformata Fouriera, dla obrazu 256x256, w 256 odcieniach szarości („Lena”) wykonana 10-krotnie;
- b) filtrowanie nieliniowe – piksel przyjmuje wartość kurtozy wyliczonej dla otoczenia 9x9;
- c) filtrowanie nieliniowe, medianą (dla otoczenia 9x9);

- d) operacje morfologiczne – para tych operacji (erozja i dylatacja) wykonywana była 10 razy;
- e) filtracja splotowa (filtr uśredniający 9x9).

Tabela 7

Czasy realizacji programów

| Typ testu | Khoros | Classman | Matlab                       |
|-----------|--------|----------|------------------------------|
| a         | 28s    |          | 8,1s                         |
| b         | 6,5s   | 26,1s    | 607,48s/485,22s <sup>6</sup> |
| c         | 3s     | 7,8s     | 6,04s                        |
| d         | 3,4s   | 41,7s    | 9,61s                        |
| e         | 3s     | 9,3s     | 2,14s                        |

Ze względu na to, iż pakiety nie pracują w tych samych środowiskach operacyjnych, wspólnym elementem testów był jedynie obraz („Lena” o wymiarach 256x256 i 256 kolorach) i komputer, na którym przeprowadzano testy. Był to komputer z procesorem ADM K5 – 166 MHz pod kontrolą systemu operacyjnego Red Hat Linux 5.0 (2.0.32) oraz systemem MS Windows 95.

Dla pakietu Matlab filtrowanie kurtozą przeprowadzono na dwa sposoby. Oba wykorzystywały standardową funkcję nlfiter (filtrowanie nieliniowe), z tym że: pierwsza stosowała kurtozę zdefiniowaną w m-pliku; a druga w pliku mex.

Dla Khorosa analogiczna operacja została zaimplementowana w całości w języku C.

Różnica czasu przeprowadzania operacji testowych pomiędzy różnymi platformami wynikała, w znacznej mierze, ze sposobu implementacji tych poleceń. Szybciej wykonywane były operacje zaimplementowane w całości w C, wolniej częściowo realizowane z poziomu makroprogramów. Należy jednak uznać, że stanowi to odwzorowanie charakterystyk pakietów.

## 10. Podsumowanie

Najszybszym pakietem okazał się Khoros, jedynie w teście wydajności wykonania szybkiej transformaty Fouriera ustąpił on pierwsze miejsce Matlabowi.

Wszystkie pakiety są nadal rozwijane. Dla najpopularniejszych z trójki: Matlaba i Khorosa, można znaleźć wiele dodatkowych przyborników w internecie. Także przykłady rozwiązań dla wielu problemów z zakresu wizji komputerowej, prezentowane w internecie, zostały uzupełnione o programy zaimplementowane dla środowiska Matlab lub Khoros.

<sup>6</sup> Drugi pomiar, dla połączenia nlfiter i funkcji typu mex (dll).

Z drugiej strony, wydaje się, że łatwiej jest stworzyć samodzielne programy (zwłaszcza komercyjne) w oparciu o programy przygotowane dla Classmana czy Khorosa niż Matlaba. W przypadku Khorosa jest to zresztą niemal automatyczne, gdyż polecenia Khorosa są niezależnymi programami działającymi w systemie Unix.

Mamy nadzieję, że wykonane porównanie platform programowych ułatwi nam wybór środowiska pracy do dalszych prac Zespołu Komputerowych Systemów Wizyjnych.

## LITERATURA

1. Drozdowski P.: Wprowadzenie do MATLAB-a, Politechnik Krakowska, Kraków 1995.
2. Gander W., Hřebíček J.: Solving Problems in Scientific Computing Using Maple and MATLAB®, Springer-Verlag, Berlin Heidelberg 1995.
3. Mrozek B., Mrozek Z.: Matlab – uniwersalne środowisko do obliczeń naukowo-technicznych – wprowadzenie do programowania, Ago, Kraków 1994.

Recenzent: Dr inż. Katarzyna Stapor

Wpłynęło do Redakcji: 8 września 1998 r.

## Abstract

There are many software platforms for computer vision. They differ in features, availability, development principles, usability and efficiency.

Some of them were developed in research centers, for example: Khoros (from the University of New Mexico, in Albuquerque), Classman (from Centre d'Automatique de Lille) or Xite (from University of Oslo).

Other platforms are derived from popular general-purpose computing platforms like Matlab, which is derived from a language for technical computing (especially for matrix analysis and manipulation).

There are some platforms dedicated to special problems, like Midas, which is a medical image processing package.

We present three software platform used in computer vision: Khoros, Matlab (with the Image Processing toolbox) and Classman. All of them were used and tested by the Computer Vision Systems Group.



These programs were written, for different operating systems: Classman and the tested version of Matlab work under MS Windows, while Khoros runs under Unix.

We describe their features for computer vision; especially: types of image files, forms of images, colours operations, filters, histogram operations, edge detectors, morphology operations, transforms and statistical features.

We also describe development principles – how users can write their own programs used with those packages.

Some performance tests were written. We tested, how much time is needed for typical operations like: fast Fourier transform, convolutions, non-linear filtering or morphological operations.

For most operations, Khoros proved to be fastest of the three programs. For fast Fourier transform the best performer is Matlab.

These tests and feature comparisons help us to choose one of the packages for future works in Computer Vision Systems Group.