

СЕРЕДОВИЩЕ «ШВИДКОЇ» РОЗРОБКИ СИСТЕМ КОМП'ЮТЕРНОЇ ОБРОБКИ ЗОБРАЖЕНЬ

В роботі запропоновано підхід до декомпозиції задач комп'ютерної обробки зображень на основі використання проміжних семантичних моделей, розроблено предметно-орієнтовану мову опису процесу обробки зображень та реалізовано відповідний програмний інструментарій для «швидкої» розробки систем комп'ютерної обробки зображень. Розроблена система застосована для експериментального дослідження доцільності використання спеціалізованих методів для обробки зображень тексту.

Presents an approach to the decomposition of tasks of computer image processing based on the use of intermediate semantic model, developed object-oriented language describing the process of image processing and is implemented as software tools for "fast" development of computer image processing in this case. The system used for the experimental study of the feasibility of using specialized methods for image processing text.

Вступ

Розпізнавання тексту, обробка космічних знімків, машинний зір, виокремлення різноманітних характеристик, ідентифікація облич, виявлення форми та переміщення об'єктів – ось далеко неповний перелік напрямків використання комп'ютерної обробки зображень в прикладних та наукових цілях.

Сьогодні замовники прикладних інформаційних систем все частіше вимагають в якості додаткової функціональності можливість використання обробки зображень для вирішення власних прикладних задач. Наприклад, додатковою вимогою до підсистем аутентифікації може бути можливість розпізнавання користувачів за зображенням обличчя або відбитків пальців.

Використання існуючих систем та бібліотек для комп'ютерної обробки зображень для розробки такої функціональності викликає низку труднощів.

Так, наприклад такі системи, як Adobe FineReader[1], Readiris[2] та Nuance OmniPage[3], є закритими і комерційними, тому не можуть бути адаптовані для вирішення прикладних задач замовника, або потребують значних фінансових затрат, пов'язаних з придбанням ліцензій на їх використання. Бібліотеки, наприклад такі, як OpenCV[4], Graphics Magick[5], Image Processing Toolbox[6] та GEGl[7], носять універсальний характер і надають розробнику програмного забезпечення засоби «низького» рівня та містять обмежену кількість прикладів використання і вкрай низькоякісну документацію. Це збільшує трудоміс-

ткість процесу розробки підсистем обробки зображень (ПОЗ), завдяки чому розробники опиняються в ситуації, коли вартість реалізації додаткової функціональності системи сумірна або перевищує загальну вартість проекту, яка влаштовує замовника.

В зв'язку з цим, **метою роботи** є зниження трудомісткості розробки ПОЗ за рахунок реалізації та використання засобів «швидкої» розробки відповідного призначення.

Для досягнення цієї мети треба вирішити проблему спрощення розв'язку задач обробки зображень за рахунок декомпозиції перетворень моделей зображень, розробити спосіб параметричного налаштування узагальненого обчислювального процесу, що реалізує обробку зображень, реалізувати відповідні програмні засоби «швидкої» розробки та експериментально перевірити їх працездатність.

Класифікація моделей та задач комп'ютерної обробки зображень

Використання різноманітного математичного апарату, врахування при цьому в різній степені апріорної інформації про зображення не дозволяють виокремити універсальну модель зображення, яка конструктивно може використовуватись для вирішення всіх відомих задач обробки зображень.

Так особливість *сигнальної моделі* [8] полягає в тому, що в ній апріорні відомості про характер та семантику візуальної інформації враховуються найменше, що дозволяє реєструвати та зберігати будь-яку візуальну інформацію в єдиній формі.

В цьому випадку зображення подане як просторовий розподіл деякої фізичної величини, яка, зазвичай, асоціюється з яскравістю:

$$a(t): T \rightarrow I, T \subseteq X \times Y,$$

де T є площиною зображення, визначеній в просторових координатах X, Y , а I – область значень яскравості.

В залежності від того, яким чином визначені множини X, Y і I сигнальні моделі можна розділити на *неперервні* (за яскравістю та простором) та *дискретні*.

Якщо при реєстрації зображень за допомогою спеціальних пристроїв перевага надається неперервним моделям, то комп'ютерна обробка зображень оснований на використанні дискретних моделей.

Зазвичай зображення як сигнал розглядається у вигляді вектора в ортонормованому функціональному базисі [9]:

$$\begin{aligned} \Phi &= \{\phi_k(t)\}, \\ \phi_k(t) &: T \rightarrow I, \\ \forall i, j &: (\phi_i(t)\phi_j(t)) = 0, \\ a(t) &= \sum_k \alpha_k \phi_k(t). \end{aligned}$$

Функції $\phi_k(t)$ визначені на тих самих множинах T і I , і зображення може розглядатися як елемент лінійного простору $\phi_1(t) \times \phi_2(t) \times \dots \times \phi_k(t)$, а його обробка – як лінійні (нелінійні) перетворення в цьому просторі.

Якщо множина базисних функцій Φ є скінченною, то зображення можна подати як скінченну лінійну комбінацію за допомогою набору коефіцієнтів $\langle \alpha_1, \alpha_2, \dots, \alpha_k \rangle$ його розкладу в базисі Φ .

Такий принцип покладено в основу дискретного подання зображень, коли зображення розкладається в базисі дельта-функцій відліків та подається у матричному вигляді. У протилежність до цього, неперервні моделі оснований на використанні нескінченного базису, в якому лінійна комбінація визначається як:

$$a(t) = \int_F \alpha(f) \phi(f, t) df.$$

Перехід від неперервної сигнальної моделі до дискретної здійснюється за допомогою перетворення:

$$a_{i,j} = \int_{-\infty}^{+\infty} a(t) \delta(t - \frac{k}{2F}) dt.$$

Математичний апарат, який використовується для побудови сигнальних моделей, є зручним для опису перетворень, які складають сутність задач корекції та попередньої обробки зображень, пов'язаних з фільтрацією. При цьому використовується інтегральна згортка з маскою (імпульсний відгук) фільтру h [9]:

$$b(t) = \int_{-\infty}^{+\infty} a(\tau) h(t - \tau) d\tau,$$

$$b_{k,l} = \sum_{i=-n}^n \sum_{j=-n}^n a_{k-i,l-j} h_{i,j}.$$

Однак при вирішенні задач розпізнавання зображень використовуються інші, семантичні моделі зображень [8], які на відміну від сигнальних моделей, містять семантику зображених об'єктів в тому, чи іншому вигляді.

Семантичні моделі представляють собою опис в термінах деякої множини елементів O та структур a , заданих на цій множині:

$$\begin{aligned} a &= \{S_i\}, \\ S_i &\subseteq O \times O \times \dots \times O, \\ O &= \{o_k\}, \\ o_k &\in P_1 \times P_2 \times \dots \times P_m. \end{aligned}$$

Перехід від сигнальної моделі зображення до семантичної складає сутність задачі розпізнавання зображень, при цьому існує низка підходів до її вирішення.

Один з підходів передбачає порівняння зображення з еталоном, які можуть бути подані не лише сигнальною моделлю, а й набором деяких узагальнених параметрів, потужність якого може бути набагато меншою, ніж кількість відкликів матричної моделі зображення. Ступінь співпадіння з еталоном оцінюється відстанню між зображенням і еталоном у відповідному параметричному просторі.

Отримання значень параметрів пов'язано з їх статистичною обробкою, що приводить до статистичної трактовки семантичної моделі зображення, коли зображенню в цілому, або його окремим фрагментам у відповідність поставлені їх статистичні характеристики. Такий підхід, коли зображення розглядається як «чорна скринька», дозволяє, наприклад, доволі ефективно виділяти фон на зображеннях, будувати байєсівські класифікатори для піксельного розпізнавання зображень тощо.

Але в тому випадку, коли семантика зображення пов'язана саме з геометрією зображених

об'єктів, використовують моделі, що активно використовують апріорну інформацію про тип зображення, який зазвичай можна задати системою правил побудови, виражених в термінах геометричних елементів та структур, визначених на них.

Перехід від одної семантичної моделі до іншої, так само, як перехід від сигнальної моделі до семантичної, пов'язаний з вирішенням задачі розпізнавання. Таким чином, задачу розпізнавання зображення в загальному вигляді можна розбити на послідовність менш складних задач (переходів) з використанням проміжних семантичних моделей.

Сутність таких переходів полягає в тому, що у відповідність певному відношенню на множині елементів однієї моделі ставиться поняття та відповідний їй елемент іншої моделі.

Нехай E_i – множина елементів моделі i -ого рівня, тоді перехід від моделі $i-1$ -ого до моделі i -ого рівня пов'язаний з пошуком однозначного відображення $v: S(E_{i-1}) \rightarrow E_i$, де

$$S(E_{i-1}) \subseteq E_{i-1} \times E_{i-1} \times E_{i-1} \dots \times E_{i-1}.$$

Відображення v визначає процес розпізнавання множини елементів, які розглядаються як елементи простору E_{i-1}^t .

Розмірність простору t відповідає складності правил формування моделі зображення i -ого рівня.

Таким чином, загальна методика вирішення задач розпізнавання зображень полягає у виборі цільової семантичної моделі, низки проміжних семантичних моделей, процедур, які реалізують перетворення цих моделей, процедур, які реалізують попередню обробку зображень на рівні сигнальної моделі та способу переходу від сигнальної моделі до відповідної семантичної моделі.

Переваги такого підходу полягають в тому, що декомпозиція загальної задачі на послідовність менш складних задач суттєво знижує трудомісткість вирішення за рахунок зниження розмірності параметричного опису об'єктів розпізнавання. При цьому для вирішення окремих задач можна використовувати різні моделі зображень і відповідні їм методи.

В рамках такого підходу можна запропонувати узагальнену структуру параметризованого обчислювального процесу для вирішення задач комп'ютерної обробки зображень на основі використання розширюваної бібліотеки процесів перетворення моделей зображень, що створює передумови для розробки системи

комп'ютерної обробки зображень, здатної генерувати модулі відповідного призначення для використання в інших прикладних інформаційних системах.

Предметно-орієнтована мова для вирішення задач комп'ютерної обробки зображень

Мета розробки предметно-орієнтованої мови полягає у наданні розробникам інформаційних систем засобів для параметричного налаштування узагальненого обчислювального процесу комп'ютерної обробки зображень, а також процесу кодогенерації відповідної підсистеми обробки зображень (ПОЗ), використання яких не передбачає детального знання як методів обробки зображень, так і специфікації внутрішніх класів самої системи обробки зображень.

Розробка нової предметно-орієнтованої мови складається з етапу створення абстрактного синтаксису, на якому визначається множина семантичних одиниць, етапу розробки синтаксису, на якому формально визначається подання семантичних одиниць синтаксичними, графічними або змішаними засобами, та етапу розробки правил трансформації абстрактного подання в таке, що виконується інструментальним оточенням.

При формулюванні множини понять, пов'язаних з розробкою мови та реалізацією інструментальних засобів використано мову UML та термінологію структурних шаблонів проектування [10, 11].

На рис.1 подано загальну схему відношень між семантичними одиницями предметно-орієнтованої мови.

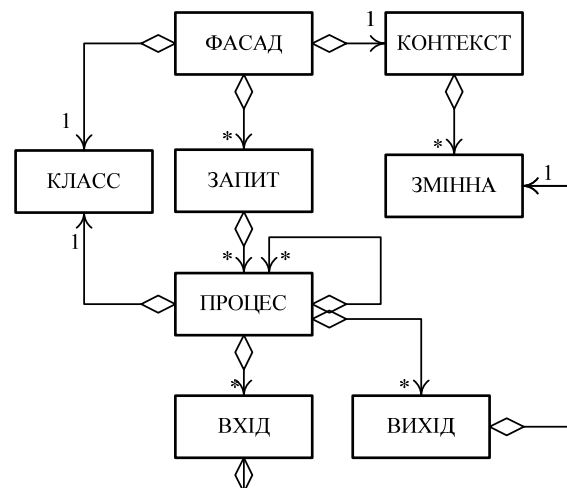


Рис.1. Схема відношень між семантичними одиницями предметно-орієнтованої мови комп'ютерної обробки зображень.

З точки зору розробника прикладної інформаційної системи підсистема обробки зображень (ПОЗ) повинна бути подана фасадом, в межах якого специфікована множина запитів до ПОЗ. Необхідна інформація для генератора вихідного коду ПОЗ містить множину ідентифікаторів цих запитів та ідентифікатор класу фасаду. Взаємодія між основною системою та ПОЗ здійснюється за рахунок викликів методів-запитів з боку основної системи та доступу до глобального контексту ПОЗ, до якого входять задекларовані змінні, які ідентифікуються за ім'ям.

Окремий запит подано, як послідовність параметрично налагоджених процесів, для кожного з яких вказано клас реалізації. Сам процес реалізує окрему задачу перетворення моделей зображень.

Налагодження процесу відбувається завдяки визначенню його входів та виходів, для яких вказана прив'язка до змінних глобального контексту ПОЗ. Також для входів можна використовувати значення простих типів: ціле або дійсне значення, логічне значення, текстове значення.

Для виконання одного запиту ПОЗ може виникнути необхідність виконання іншого запиту. З цією метою використано залежності між процесами, які на їх множині утворюють композиційну структуру.

Зрозуміло, що наведений опис носить декларативний характер і може бути поданий за допомогою мови XML[12]. Приклад такого опису приведено нижче:

```
<fasade name="TestImageIO">
<include src = "src/xml/lib/image.xml"/>
<var name="image"/>
<var name="gs"/>

<request name="LoadImage"
description="Load Image from file">
<process type="image.load">
<input name="src" value =
"res/images/flipVertical.png"/>
<output name="dest" ref="image"/>
</process>
</request>

<request name="ConvertImage"
depends="LoadImage"
description = "Convert Image to
grayscale">
<process type="image.toGrayScale">
<input name="src" ref="image"/>
<output name="dest" ref="gs"/>
</process>
</request>
```

```
<request name="SaveImage"
depends="ConvertImage"
description="Writes image into file">
<process type="image.save">
<input name="src" ref="image"/>
<input name="dest" value =
"res/log/alpha_c.png"/>
</process>
<process type="image.save">
<input name="src" ref="gs"/>
<input name="dest" value =
"res/log/alpha_gs.png"/>
</process>
</request>
</fasade>
```

Цей приклад описує типову ПОЗ для відлагодження реалізації процесів, фасад якої буде подано класом TestImageIO, що інкапсулює запити:

```
public void LoadImage();
public void ConvertImage ();
public void SaveImage();
```

та методи доступу до змінних глобального контексту ПОЗ:

```
public Object getImage();
public void setImage(Object image);
public Object getGs();
public void setGs(Object gs);
```

Прив'язка типів процесів до класів реалізації здійснюється в окремому XML-описі, що міститься в файлі image.xml та завантажується за допомогою інструкції include. Цей файл містить прив'язку імен типів до класів реалізації процесів:

```
<inclusion>
<classpath url = "file://bin/" />
<processdef type="image.load"
implementation =
"process.app.image.LoadImageProcess" />
<processdef type="image.save"
implementation =
"process.app.image.SaveImageProcess" />
<processdef type =
"image.toGrayScale"
implementation = "
app.image.ToGrayScaleImageProcess" />
</inclusion>
```

В якості графічної форми подання процесу обробки зображень, яка здійснюється ПОЗ можна використати стандартні схеми опису процесу обчислень з специфікацією даних та процедур. Приклад такого опису приведений на рис.2.

Інструментальні засоби реалізації запропонованої предметно-орієнтованої мови комп'ютерної обробки зображень розроблені з

використанням мови програмування Java та стандартного пакету математичних бібліотек з відкритим кодом Apache Common Math [13]. Основними класами є інтерпретатор мови, який дозволяє реалізувати режим відлагодження ПОЗ. В результаті розбору XML-файлу за допомогою SAX-парсера [14] здійснюється побудова глобального контексту ПОЗ, завантаження класів-реалізацій процесів та побудова ланцюгів виконання.

Процес виконання складається з фази перевірки відповідності налагодження процесу за допомогою входів і виходів специфікації процесу, яку інтерпретатор отримує під час виконання, та стадії виконання процесу.

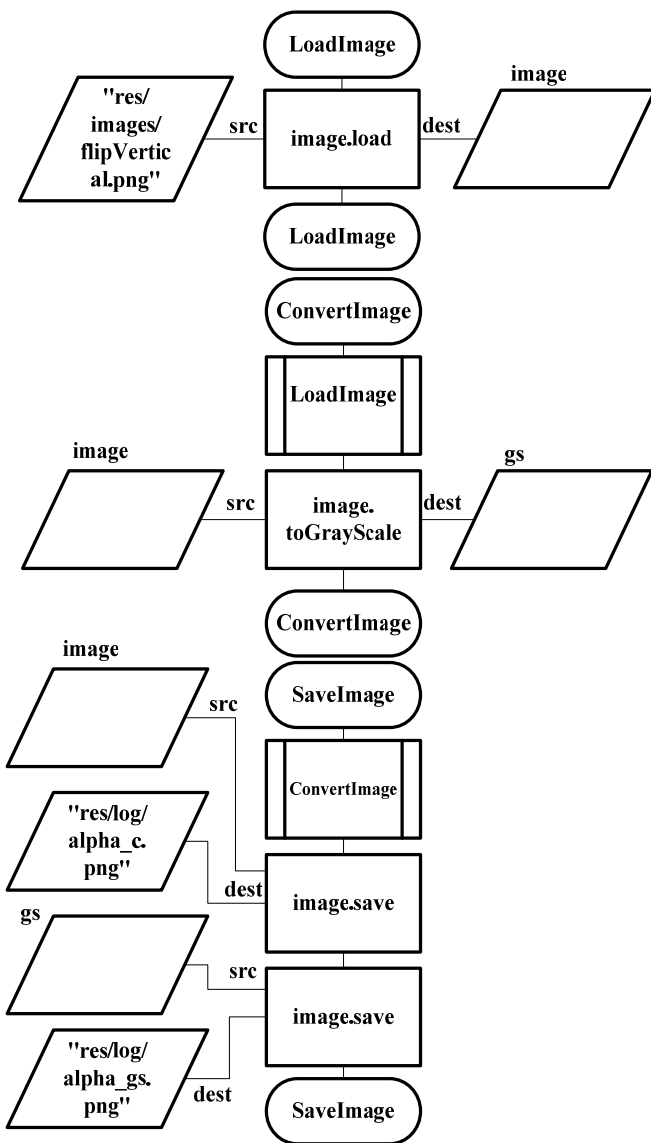


Рис.2. Приклад графічного опису ПОЗ.

Сама ж специфікація процесу здійснюється розробником процесу за допомогою механізму анотування ресурсів, який реалізовано в мові Java.

Нижче наведено приклад такої специфікації:

```

...
@description ("Save image to file")
@inputList({
    @input (
        description = "image object",
        optional = false,
        name = "src",
        type = BufferedImage.class
    ),
    @input (
        description = "image filename",
        name = "dest",
        type = String.class
    )
})
public class SaveImageProcess extends
TaskImpl {...

```

Використання таких анотацій дозволяє формалізувати процес перевірки коректності налаштування процесу та спростити розробку класів-реалізацій процесів.

Також активне використання інформації часу виконання дозволяє створити середовище швидкої розробки ПОЗ, наприклад у вигляді розширень для середовищ Eclipse [15] та Idea [16], підключати до такого середовища додаткові модулі, що містять класи-реалізації процесів та отримувати з них всю необхідну для їх використання інформацію.

Розроблена система була випробувана для виявлення доцільності використання векторно-пиксельної моделі та методу деформації координатної сітки [17] для обробки зображень тексту.

З цією метою для вихідного зображення тексту була побудована векторно-пиксельна модель, з її допомогою виконано 10-кратне збільшення зображення та деформація координатної сітки. Результати обробки зображені на рис.3. в масштабі 25% від оригінального розміру.

Мета обробки полягає у зменшенні кількості артефактів при багатократному збільшенні масштабу зображень тексту. Як видно на рис.3.а., мають місце численні артефакти на границях символів, обумовлені загубленням сигнальної моделі в результаті rasterization. Як видно з рис.3.б., сумарна площа таких артефактів може бути суттєво зменшена за рахунок використання методу деформації координатної сітки.

Висновки

Запропоновано підхід до декомпозиції задачі обробки зображень на основі використання проміжних семантичних моделей. Це дозволило визначити узагальнену структуру відповідного

обчислювального процесу та спосіб декларативного опису його налаштування.

Запропонована предметно-орієнтована мова та реалізовані відповідні інструментальні засоби, які складають основу середовища «швидкої» розробки ПОЗ.

В подальшому планується розширення набору бібліотек, які реалізують методи комп'ютерної обробки зображень, та використання розробленого середовища в наукових дослідженнях та в навчальному процесі.

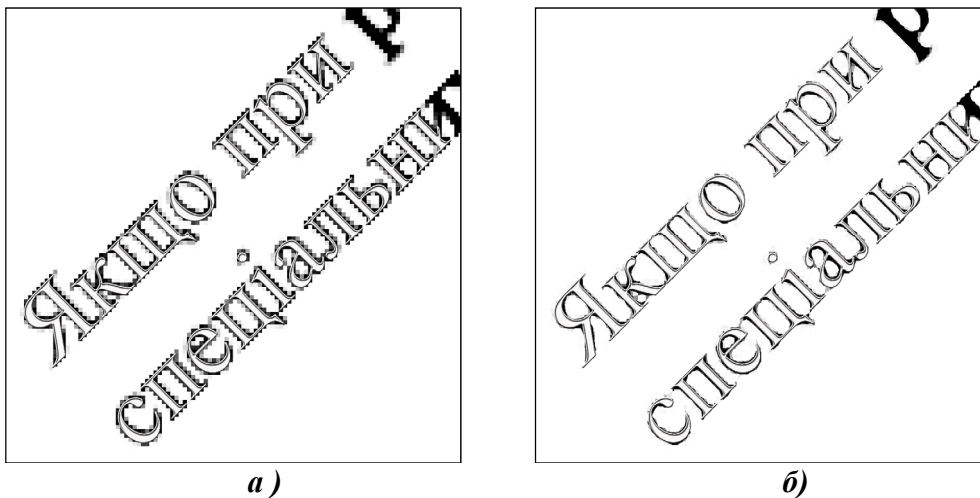


Рис.3. Результати обробки зображення тексту: а) без використання деформації координатної сітки; б) з використанням деформації координатної сітки.

Список літератури

1. Abbyy FineReader [Електроний ресурс] / Компанія АBBYY. – 2012. – Режим доступу: <http://www.abbyy.ua/products/ocr/finereader/>.
2. Readiris [Електроний ресурс] / Компанія I.R.I.S. s.a. – 2012. – Режим доступу: <http://www.irislink.com/c2-2115-189/Readiris-14--OCR-Software--Scan--Convert---Manage-your-Documents-.aspx>.
3. Nuance OmniPage [Електроний ресурс] / Компанія Nuance Communications, Inc. – 2012. – Режим доступу: <http://www.nuance.com/for-business/by-product/omnipage/professional/index.htm>.
4. OpenCV [Електроний ресурс] / Компанія Itseez. – 2012. – Режим доступу: <http://opencv.org/>.
5. GraphicsMagick [Електроний ресурс] / GraphicsMagick Group. – 2012. – Режим доступу: <http://www.graphicsmagick.org/>.
6. Image Processing Toolbox [Електроний ресурс] / Компанія The MathWorks, Inc. – 2012. – Режим доступу: <http://www.mathworks.com/products/image/>.
7. GEGL (Generic Graphics Library) [Електроний ресурс] / Компанія GIMP. – 2012. – Режим доступу: <http://www.gegl.org/>.
8. Бузовский О.В. Компьютерная обработка изображений / Бузовский О.В., Болдак А.А., Мохамед Руми М.Х. – К.: Корнійчук, 2001. – 180 с.
9. Методы компьютерной обработки изображений / [ред. Сойфер В.А.]. – [2-е изд.]. – М.: ФИЗМАТЛИТ, 2003. – 784 с.
10. Буч Г. UML. Классика CS / Буч Г., Якобсон А., Рамбо Дж.; пер. с англ.; под общей редакцией проф. С. Орлова. – [2-е изд.]. – СПб.: Питер, 2006. – 736 с.
11. Приемы объектно-ориентированного проектирования. Паттерны проектирования / [Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж.]; пер. с англ. – [2-е изд.]. – СПб.: Питер, 2001. – 368 с.
12. Extensible Markup Language (XML) 1.0 (Fifth Edition) [Електроний ресурс] / World Wide Web Consortium. – 2012. – Режим доступу: <http://www.w3.org/TR/REC-xml/>.
13. Commons Math: The Apache Commons Mathematics Library [Електроний ресурс] / The Apache Software Foundation. – 2012. – Режим доступу: <http://commons.apache.org/math/>.
14. D. Brownell. SAX2 / D. Brownell. – Sebastopol: O'Reilly, 2002. – 228 p.
15. Eclipse [Електроний ресурс] / The Eclipse Foundation. – 2012. – Режим доступу: <http://eclipse.org/>.
16. Idea [Електроний ресурс] / Компанія JetBrains. – 2012. – Режим доступу: <http://www.jetbrains.com/idea/>.
17. Бузовский О.В. Повышение контрастности изображения / Бузовский О.В., Невдащенко М.В., Болдак А.А. – 3б. текстів виступів на IV Всеукр. наук.-практ. конф. "Інформатика та системні науки" / Полтавський університет економіки і торгівлі. – П.: ПУЕТ, 2012. – с.31-34.