**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«КРИВОРІЗЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ»**

**Фізико-математичний факультет**

**Кафедра інформатики та прикладної математики**

|  |  |
| --- | --- |
| «Допущено до захисту»  Завідувач кафедри  \_\_\_\_\_\_\_\_\_\_\_\_ Соловйов В. М.  (підпис)  «\_\_\_» \_\_\_\_\_\_\_\_\_201\_\_\_ р. | Реєстраційний № \_\_\_\_\_\_\_  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 201\_\_ р. |

**РОЗРОБКА ВІДЕОПЕРА ДЛЯ ДОПОВІДАЧА НА НАУКОВІЙ КОНФЕРЕНЦІЇ**

Кваліфікаційна робота студента

групи Ім-12

Гузієка Олександра Васильовича,

ступінь вищої освіти - магістр

спеціальності 014.09 - вища

освіта (Інформатика).

Науковий керівник

доктор фізико-математичних наук,

професор Міненко П. О.

Оцінка:

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Шкала ECTS \_\_\_\_\_ Кількість балів \_\_\_\_\_

Голова ЕК \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище, ініціали)

Члени ЕК \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище, ініціали)

\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище, ініціали)

\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище, ініціали)

\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище, ініціали)

Кривий Ріг – 2017

ЗМІСТ

[ВСТУП 3](#_Toc503771801)

[РОЗДІЛ 1 ОПЕРАТИВНІ ЗАСОБИ СТВОРЕННЯ ВЕКТОРНОЇ ГРАФІКИ ТА ЇХНЄ ЗАСТОСУВАННЯ 5](#_Toc503771802)

[1.1 Засоби створення векторної графіки 5](#_Toc503771803)

[1.2 Відомі методи та програми векторизації та їхні недоліки 9](#_Toc503771804)

[1.3 Наявні алгоритми трекінгу об'єктів та їхнє застосування 29](#_Toc503771805)

[1.4 Висновки до розділу 1 41](#_Toc503771806)

[РОЗДІЛ 2 ПРОЕКТУВАННЯ ВІДЕОПЕРА 42](#_Toc503771807)

[2.1 Алгоритм програмного забезпечення роботи відеопера 42](#_Toc503771808)

[2.2 Інструментарій розробки відеопера 48](#_Toc503771809)

[2.3 Висновки до розділу 2 59](#_Toc503771810)

[РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ВИПРОБОВУВАННЯ ВІДЕОПЕРА 60](#_Toc503771811)

[3.1 Інтерфейс та інструкція для користувача 60](#_Toc503771812)

[3.2 Результати тестування програмної реалізації алгоритму 65](#_Toc503771813)

[3.3 Висновки до розділу 3 69](#_Toc503771814)

[ВИСНОВКИ 70](#_Toc503771815)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 71](#_Toc503771816)

# ВСТУП

**Актуальність теми**. У теперішній час обчислювальна техніка використовується в багатьох областях людської діяльності, будучи зручним і багатофункціональним інструментом вирішення широкого кола завдань. Багато галузей техніки, що мають відношення до отримання, обробки, зберігання та передачі інформації, значною мірою орієнтуються в даний час на розвиток систем, у яких інформація має характер зображень.

Разом із тим, рішення наукових та інженерних задач при роботі з візуальними даними вимагає особливих зусиль, що спираються на знання специфічних методів, оскільки традиційна ідеологія одновимірних сигналів і систем мало придатна в цих випадках. В особливій мірі це проявляється при створенні нових типів інформаційних систем, що вирішують такі проблеми, які до сьогодні в науці й техніці не вирішувалися, і які вирішуються зараз завдяки використанню інформації візуального характеру.

**Мета:** Удосконалити алгоритм і програму для супроводження дії доповідача на науковій конференції.

Для досягнення мети слід розв’язати такі задачі:

1. Проаналізувати наявні алгоритми трекінгу об’єкта.
2. На основі результатів аналізу обрати найбільш оптимальний алгоритм.
3. Обґрунтувати вибір інструментів розробки відеопера та

створити програмну реалізацію оптимального алгоритму.

1. Здійснити тестування та корекцію програмного забезпечення.

**Об’єктом дослідження** є інформаційні технології для демонстрації наукових, навчальних та рекламних проектів і матеріалів.

**Предметом дослідження** є розробка програми відеопера для доповідача на науковій конференції.

**Наукова новизна роботи.** Удосконалено алгоритм комп'ютерного зору для доповідача на науковій конференцій за рахунок доповнення алгоритму сервісними процедурами: вибору кольору пера, товщини ліній і "гумки", які запрограмовані й вставлені в програму відеопера.

**Практична значимість** роботи полягає в тому, що її результати можуть використовуватись не тільки для демонстрації наукових, учбових та рекламних матеріалів, а й для створення бізнес-проектів, систем відео спостереження, батьківського контролю, штучного інтелекту з елементами поточних вказівок.

**Структура та обсяг кваліфікаційної роботи. Р**обота складається вступу, трьох розділів, висновків та списку використаних джерел. Загальний обсяг роботи становить 73 сторінки та містить 29 рисунків.

# РОЗДІЛ 1 ОПЕРАТИВНІ ЗАСОБИ СТВОРЕННЯ ВЕКТОРНОЇ ГРАФІКИ ТА ЇХНЄ ЗАСТОСУВАННЯ

## 1.1 Засоби створення векторної графіки

Векторна графіка  (геометричне моделювання або об'єктно-орієнтована графіка) – це створення зображення в комп'ютерній графіці із сукупності геометричних примітивів  (точок, ліній, кривих, полігонів), тобто об'єктів, які можна описати математичними виразами.

Сучасні дисплеї можна вважати плоскою мережею точок, які можуть бути пофарбовані в різні кольори. Точки називаються пікселями (англ. picture elements). Чим менше розмір точок, та чим щільніше вони розташовані, тим якіснішим буде зображення, але очевидно, що розмір файлу, який містить опис зображення буде пропорційний якості зображення.

Сучасні дисплеї та принтери — це растрові пристрої. Перед тим як відобразити, або надрукувати векторне зображення, спочатку треба його перетворити в растрове зображення — масив пікселів. Розмір створюваного растрового зображення залежить від використаної роздільної здатності растрового пристрою. Таким чином, легко перевести векторне зображення у відповідний растровий формат, а зворотне перетворення дуже складне. Зображення переведене з векторного формату в растровий збільшується в розмірах, та втрачає властивість масштабування без втрати роздільної здатності. Також втрачається можливість редагувати елементи зображення як окремі об'єкти. Розмір векторного зображення залежить від кількості елементів зображення, та від переліку їх властивостей.

У комп'ютерній типографії сучасні шрифти описуються алгебраїчними кривими другого або третього степеня з контрольними точками. Також використовуються растрові шрифти. Отримання растрового шрифту по векторному опису — не тривіальна задача. Наприклад, треба уникнути «зубчатості».

На початку комп'ютерної епохи в 1950 році, а також в 1980, використовувались різні типи відображення векторної графічної системи. У цих системах електронне ядро ЕПТ монітора направлялось прямо щоб намітити необхідну форму, лінійний сегмент як лінійний сегмент, залишок екрану при цьому відображається чорним. Цей процес повторювався багато разів на секунду щоб уникнути блимання картинки. Ця система дозволяє відображати лінійне зображення з дуже високою роздільною здатністю, і переміщати зображення, які є показані без (на цей час) немислимо величезної кількості пам'яті, яка була б потрібна системі растрово-еквівалентного рішення. Ці засновані на векторі монітори були також відомі як X-Y displays.

Векторна графіка описує зображення з використанням відрізків прямих і вигнутих ліній, які називаються векторами, а також параметрів, що описують їхні кольори та розташування. Наприклад, зображення деревного листа описується точками, через які проходить лінія, створюючи, тим самим, контур листа. Колір листа задається кольором контуру та області всередині цього контуру.

При редагуванні елементів векторної графіки змінюються параметри прямих і вигнутих ліній, що описують форму цих елементів. Можна переносити елементи, міняти їхній розмір, форму й колір, але це не відіб'ється на якості їх візуального представлення. Векторна графіка не залежить від дозволу, тобто вона може бути показана в різноманітних вихідних пристроях із різним дозволом без втрати якості.

Векторне подання полягає в описі елементів зображення математичними кривими з їхніх кольорів і заповнюваності (треба згадати, що коло й окружність – це різні фігури). Червоний еліпс на білому тлі буде описаний лише двома математичними формулами - прямокутника й еліпса відповідних кольорів, розмірів і розташування. Очевидно, таке опис займе значно менше місця, ніж у першому випадку. Ще одна перевага – це якісне масштабування в будь-яку сторону. Збільшення або зменшення об'єктів досягається збільшенням або зменшенням відповідних коефіцієнтів у математичних формулах. На жаль, векторний формат стає невигідним при передачі зображень із великою кількістю відтінків або дрібних деталей (наприклад, фотографій). Адже кожен дрібний відблиск у цьому випадку буде представлятися не сукупністю одноколірних точок, а найскладнішою математичною формулою або сукупністю графічних примітивів, кожен з яких, є формулою. Це призводить до обваження файлу. Крім того, переклад зображення з растрового у векторний формат (наприклад, програмою Adobe Strime Line або Corel OCR-TRACE) призводить до спадкоємства останнім неможливості коректного масштабування в більшу сторону. Від збільшення лінійних розмірів кількість деталей або відтінків на одиницю площі більше не стає. Це обмеження накладається дозволом вступних пристроїв (сканерів, цифрових фотокамер і ін.).

У даний час створено безліч пакетів ілюстративної графіки, які містять прості в застосуванні, розвинені й потужні інструментальні засоби векторної графіки, призначеної як для підготовки матеріалів до друку, так і для створення сторінок в Інтернеті.

Для створення графічного об'єкта потрібна програма ілюстративної векторної графіки. Якість і корисність засобів векторної графіки визначаються, головним чином, можливостями масштабування.

Пакети векторної або ілюстративної графіки завжди грунтувалися на об'єктно-орієнтованому підході, що дозволяє малювати контури об'єктів, а потім зафарбовувати їх або заповнювати візерунками. Можна дуже точно відтворювати ці контури, задаючи будь-який розмір, оскільки вони формуються за допомогою математичної моделі з точок і кривих, а не як растрові зображення - у вигляді сітки, заповненої прямокутними пікселями.

До числа нових можливостей, виявлених у цій категорії виробів, відноситься миттєве градієнтне зафарбування. Такі примітиви, як багатокутники, зірки та спіралі, стали звичайними атрибутами подібних пакетів. Пов'язані кольори дозволяють замінити червоний колір троянди на жовтий, змінивши лише базовий колір; усі пов'язані відтінки зміняться автоматично. Багатошарові інтерактивні кольорові "діапозитиви" забезпечують раніше недосяжну глибину. І можна перетворювати векторні зображення в растрові в рамках векторного графічного файлу. Якщо вчорашні пакети векторної графіки дозволяли лише поміщати растрове зображення в наш файл, то за допомогою сучасних програм можна вбудовувати представлені в растровій формі зображення, змінювати їхні розміри й навіть накладати спеціальні ефекти й маски. Це полегшує процес отримання остаточного зображення засобами багатошарової графіки - об'єднанням векторних і растрових файлів необхідним, для створення логотипів, друкованих рекламних оголошень і картинок для Web.

З усіх нововведень найбільше зацікавлення представляють фірмові зовнішні модулі (plug-ins) Web, які збагачують Мережу засобами векторної графіки й навігації. Файл векторної графіки незрівнянно менше растрового файлу для такого ж зображення. І це дозволяє збільшувати масштаб фрагментів зображення до 25 тис. відсотків. Тепер з'явилася можливість призначати URL (уніфікований покажчик ресурсу) будь-якого об'єкта.

Принципи, що лежать в основі останніх пакетів, повністю змінюють ставлення до векторної графіки. CorelXara 1.5 реалізує якісно новий підхід до візуалізації, має у своєму розпорядженні приголомшливими засобами створення вихідних файлів \*.GIF і \*.JPEG і феноменально швидким зовнішнім модулем браузера для роботи з векторною графікою. Пакет Expression 1.0 фірми Fractal Design дозволяє будувати контури з інших складних векторних графічних зображень, надаючи в розпорядження користувача нескінченну розмаїтість візуальних можливостей, недосяжну за допомогою інших програм.

На відміну від призначеного для початківців-користувачів програмного забезпечення настільних видавничих систем або програм редагування фото зображень, де, як правило, містяться найбільш часто використовувані засоби редагування, графічні пакети для новачків зазвичай орієнтовані на рішення конкретних завдань, наприклад, побудова діаграм або технічне креслення. Набути навичок вільного малювання кривих Безьє важко навіть для професіонала. Проте складно освоїти й основні принципи машинного креслення, наприклад, зображення розрізів і перерізів. Крім того, багато починаючих користувачів не відчувають відмінностей між растровою та векторною графікою й можуть не знати, у яких випадках і якими пакетами користуватися. Із цих причин початківці повинні узгоджувати свої завдання з можливостями програми, і переходити до повно функціонального пакету малювання, тільки коли будуть готові до цього.

У більшості випадків для створення простих ілюстрацій початківцям досить уміти працювати з тими програмними засобами, які, можливо, у них уже є. Комплекти програм Microsoft, Corel і Lotus містять інструменти малювання у своїх модулях текстового процесора.

1.2 Відомі методи та програми векторизації та їхні недоліки  
  
 Нині створено безліч пакетів ілюстративної графіки, які містять прості в застосуванні, розвинені і потужні інструментальні засоби векторної графіки, призначеної як для підготовки матеріалів до друку, так і для створення сторінок інтернет-сайтів. Для створення графічного об'єкта потрібна програма ілюстративної векторної графіки. Якість і корисність векторної графіки визначаються, головним чином, можливостями масштабування. Пакети векторної або ілюстративної графіки завжди грунтувалися на об'єктно-орієнтованому підході, що дозволяє малювати контури об'єктів, а потім зафарбовувати їх або заповнювати візерунками. Пакет CorelDraw завжди справляє сильне враження. У комплект фірма Corel включила безліч програм, у тому числі Corel Photo-Paint. Новий пакет має, безперечно, найпотужнішим інструментарієм серед усіх програм огляду, а при цьому в порівнянні з попередньою версією інтерфейс став простіше, а інструментальні засоби малювання й редагування вузлів - гнучкішими. Однак, що стосується нових функцій, зокрема, підготовки публікацій для Web, то тут CorelDraw поступається перед CorelXara.

Робота CorelDraw із кольорами CMYK залишає бажати кращого. Кольори файлів GIF і JPEG помітно відрізнялися від кольорів, що виводяться для пробного відбитка Matchprint, у той час як пакет FreeHand відтворював однакові кольори на екрані, у файлах Web і на обох принтерах.

За допомогою інструментального засобу згладженої екструзії нам без праці вдалося побудувати тривимірне зображення гітари із двовимірного контуру. Ми призначили інтерактивні "двоточкові" градієнтні діапозитиви растрових зображень і простих векторних об'єктів: для кожної із двох зазначених точок задається ступінь прозорості у відсотках. Ця технологія була запозичена з CorelXara 1.0, проте CorelDraw не дозволяє вказувати більше двох значень прозорості для кожного об'єкта або групи, у той час як за допомогою CorelXara 1.5 можна ставити в два рази більше значень прозорості.

Художні можливості оформлення тексту в CorelDraw бездоганні, а прийняті за умовчанням параметри для міжбуквених інтервалів при розміщенні тексту вздовж кривої не вимагають настройки, що виключає накладення букв, на відміну від Canvas і FreeHand. Інструмент "лупа" не має собі рівних - він дозволяє отримати безліч спеціальних ефектів, у тому числі можливість збільшення тільки фрагмента зображення й автоматичної настройки квітів тексту залежно від кольору фону.

Можна вирізати зображення, накладати кольорові фільтри й надавати растровим зображенням вид вигнутої сторінки, використовуючи двох- і тривимірні ефекти й зовнішні модулі PhotoShop. Коли нам потрібно було редагувати пікселі, CorelDraw автоматично перемикав нас на Corel Photo-Paint, де ми могли редагувати файл і зберігати його безпосередньо в CorelDraw. Однак, крім базових можливостей масштабування й засобів динамічного призначення розмірів, CorelDraw не містить спеціальних засобів підготовки технічних ілюстрацій, подібних Smart Mouse в Canvas або копіювання масивів в Designer.

Можливості реалізувати моделі CMYK - у цьому відношенні CorelDraw переслідують невдачі - все ще викликають побоювання, хоча програма тепер може працювати із системою управління кольором Kodak CMS. По-перше, для збереження сумісності з попередніми версіями CorelDraw потрібно щоразу при відкриванні CorelDraw в меню View відключати колір-корекцію Kodak. По-друге, якщо ваші принтери не входять до обмежений список дозволених для застосування периферійних пристроїв, немає гарантії, що завжди знайдеться відповідний груповий драйвер. Ми просто не змогли вивести кольорове зображення за допомогою принтера HP DeskJet 310C, оскільки для нього був відсутній драйвер пристрою, як і груповий драйвер RGB, який міг би його замінити. Це означає, що кольори на нашому екрані друкувалися синій - блакитним, кораловий - яскраво-рожевим, а колір стиглої сливи - фіолетовим.

Ветеран векторної графіки - пакет Illustrator - колись представляв собою видатне досягнення в цій галузі й послужив моделлю, яка лягла в основу всіх представлених у цьому огляді програм. Але з тих пір у кожному новому продукті з'являлося якесь удосконалення. Наприклад, Macromedia FreeHand краще виконує імпортування файлів EPS і AI і при цьому забезпечує високу точність передачі кольору у форматі CMYK, якою завжди відрізнявся Illustrator. CorelDraw вже давно підняла планку, передбачивши у своїх пакетах градієнтне зафарбовування, істинні шари, булеві операції та спеціальні ефекти. Canvas 5 має засобами редагування растрових зображень на рівні пікселів, а його робоча область досягає майже 140 м2, у той час як для Illustrator ця галузь не перевищує 0,2 м2. Micrografx Designer надає чудовий інструментарій для малювання. Він інтегрується з Windows-95 і Microsoft Office і містить засоби для підготовки технічних ілюстрацій, а CorelXara забезпечує справжню прозорість для векторних об'єктів і можливість вбудовування растрових зображень. У свою чергу Fractal Design Expression із допомогою інструмента Skeletal Strokes дозволяє отримувати самі незвичайні ефекти й видозмінювати зображення.

Нажаль, порівняно обмежений набір засобів пакету Illustrator не означає, що він простий у застосуванні. Ми підрахували, що для градієнтна зафарбування зображення веселки, що вимагає 5 клацань мишею в CorelDraw, у разі пакета Illustrator зажадає 67 клацань, оскільки доведеться створювати переходи для кожної пари основних кольорів. Градієнтна заливка тексту вздовж траєкторії був такий трудомістка, що вирішили модифікувати наш тестовий сценарій для Illustrator.

За допомогою інструментів малювання пакета Illustrator ми змогли отримати цілком прийнятне зображення контура гітари, але в нашому розпорядженні не було засобу градієнтного зафарбовування, екструзії ("видавлювання") і плавних сполучень для згрупованих або складних об'єктів. Зрештою, нам удалося створити перехід (blend) для градієнтного зафарбовування зображень двох гітар і скористатися іншим переходом, щоб отримати прості зображення гітар між ними. Здавалося, що Illustrator завантажив наш файл LOGO.EPS, але після того, як збереження файлу з логотипом ми не змогли знову відкрити його в Illustrator. Нам удалося вийти з положення, завантаживши цей файл у FreeHand, видаливши файл EPS, що викликав відмови, зберігши його у вигляді файлу AI а потім знову відкривши його в Illustrator. Кольори нашого вбудованого кольорового файлу TIFF для попереднього перегляду були дуже спотворені, а Illustrator не передбачає навіть такі звичайні функції, як корекція кольору, фільтри або перетворення растрового файлу. Що поставляється з пакетом Illustrator програмний модуль Adobe Separator (поряд із модулями Adobe Streamline, Adobe Type Manager і Adobe Type on Call) виконує кольороподіл CMYK.

Розміщення тексту вздовж криволінійної спрямовуючої не викликало труднощів, але Illustrator "наполягав" на збільшенні відстані між літерами, щоб виключити їхні накладення. Ми не змогли змінити інтервали між буквами, які були занадто великими, доки налаштували з допомогою Kerning Control. Програма Illustrator імпортувала файл RTF, правильно вказавши назву шрифту вгорі абзацу, але при цьому змінила шрифт на Helvetica. Крім того, Illustrator не дозволяє експортувати файли .GIF і JPEG для використання їх в Web. І хоча якість кольорового друку залишається найсильнішою стороною пакету Illustrator, можливості обробки кольорів CMYK в FreeHand сподобалися нам нітрохи не менше (крім того, одна й та ж версія FreeHand може працювати в середовищі як для Windows, так і Mac). Потрібно пам'ятати також про проблеми, що виникають при застосуванні Illustrator із графічними платами на основі S3 (Adobe попереджає користувачів про це).

Illustrator, свого часу проклав шлях іншим графічним пакетам, сьогодні відійшов на другий план. До тих пір поки Adobe серйозно не переробить його, ми рекомендуємо пошукати якийсь інший пакет. Якщо ж ви все ще працюєте в Illustrator і створеними з його допомогою файлами, розгляньте FreeHand в якості альтернативи.

Micrografx Designer 7 - приємна в застосуванні, хоч і не велика програма, легко впоралася з більшістю тестів, - також заслуговує на особливу увагу завдяки своїм чудовим засобам для технічних ілюстрацій. Designer 7 поряд з FlowCharter 7 і Picture Publisher 7 утворює ядро ​​комплекту Micrografx Graphics Suite, являє собою одну з найменш дорогих серед програм даного огляду.

Інструментальні засоби малювання пакету Designer - одні з найпростіших для освоєння й застосування. На контекстно-залежному рядку інструментарію вгорі екрана розташовані кнопки для створення й редагування ліній, кривих, зірок і навіть багатокутників із криволінійними сторонами (curvygon). Засіб екструзії додало зображенню нашої гітари глибину й особливу освітленість; стандартні засоби нахилу й повороту дозволили розмістити ці гітари в тривимірному просторі. Діалогове вікно форматування об'єкта містить незвично великі піктограми й області вибору зразків внутрішнього зафарбовування, стилю ліній і заповнень, тексту й таких характеристик об'єкта, як ім'я, розмір і розташування. На жаль, іноді діалогове вікно форматування об'єкта не показувало нашого заповнення об'єкта, що призводило до неправильного використання заповнення, прийнятого за замовчуванням, при змінах шрифту.

Текст, що розташовується вздовж криволінійної спрямовуючої, створював невидиму копію вихідної спрямовуючої. Хоча ми могли редагувати невидиму, котра спрямовує, і текст при цьому змінював своє розташування, він не "реагував" на зміни видимих ​​обрисів об'єкта. Подібно CorelXara, в Designer не передбачено вікно редагування тексту, що змушує вас редагувати його постійно в режимі повної відповідності WYSIWYG. Переміщення між шарами дуже незручні, і хоча можна користуватися кількома сторінками різного формату, для переміщення об'єктів між сторінками потрібен монтажний буфер.

Інструмент для технічного креслення дозволив нам накреслити кілька розташованих на однаковій відстані один від одного об'єктів і побачити результат перш, ніж його прийняти. Ми створили безліч точок прив'язки для легко динамічно форматується розмірних ліній. Унікальне засіб Reference Point дає можливість установлювати обмеження на відстані вздовж осей x і y і на величину кута повороту або примусово розміщувати всі об'єкти на певній відстані від певної точки.

Однак у комплект поставки пакета Designer входять цікаві растрові фільтри й ефекти, і він дозволив нам редагувати пікселі в Picture Publisher засобами технології OLE. Designer формував хороші файли GIF зі змішанням квітів, файли GIF без змішування кольорів із зображеннями, що нагадують плетені кошики, і аномальні файли JPEG із зображеннями, які мають вигляд пухирців. Designer дозволяє також приєднувати URL до об'єктів, щоб використовувати їх разом із зовнішнім модулем браузера Micrografx QuickSilver 3. Цей браузер надав нам можливість збільшувати й зменшувати масштаб зображення, але ми не могли змінити положення зображення всередині кадру, подібно до того як ми робили це за допомогою зовнішнього модуля CorelXara чи Macromedia Shockwave.

Відмітна особливість QuickSilver полягає в тому, що ви можете призначати певні властивості векторних графічних об'єктів. Ми вибрали гітару, потім у меню вказали на необхідність появи текстового вікна, коли курсор за допомогою миші опинявся на гітарі. Вибравши другий пункт меню, можна прибирати з екрану текстове вікно, як тільки курсор переміщається за межі зображення гітари. Ви можете створювати анімації, переміщуючи об'єкти по кільцю, або циклічно змінюючи об'єкт, наявний зверху. В результаті виходить сценарій, за яким будь-хто може легко створювати інтерактивні додатки Web.

Designer 7 із допомогою простого інтерфейсу дозволяє легко впоратися з багатьма типовими для офісу графічними роботами, але принципові обмеження інструментальних засобів і дуже вбогі можливості для чотириколірного друку CMYK можуть змусити професійних художників-графіків утриматися від його придбання. Але якщо вам необхідно потужний засіб для технічного креслення або ви хочете, працюючи в інтерактивному режимі, розміщувати матеріали на своїх сторінках Web, обходячись при цьому без програмування, то можливо вам слід зупинити свій вибір саме на цьому пакеті.

Пакет Macromedia FreeHand 7 вражають бездоганною якістю виведення на екран і чотириколірної друку CMYK і наявністю декількох форматів для Web. Оскільки FreeHand завжди відображає кольори так само, як вони будуть виглядати при друку, ця програма була єдиною в нашому огляді, що не допускала, до нашого великого задоволення, створення чи призначення квітів, які при друку сильно відрізнялися б від відповідних кольорів на екрані.

У списку квітів FreeHand вказуються тільки ті кольори, які були використані або створені. Програма дозволяє відбирати кольору з кількох бібліотек, у тому числі Pantone і Hexachrome для друку, і з палітри Web, оптимізованою як для Mac, так і PC.

Для додавання кольорів у наш список і присвоєння імен усім кольорам імпортованих растрових зображень ми користувалися тими, які входять у пакет засобом Xtra; подібне засіб відсутня в інших програмах. Архітектура Xtra дозволяє працювати із зовнішніми програмними модулями, що дає можливість незалежним постачальникам додавати свої засоби в меню Xtra, але не інтегрувати їх у програму. Нарешті, нам удалося знайти "колірну піпетку"; цей інструмент знаходиться на панелі інструментарію Xtra Tools, а не на панелі змішання квітів Color Mixer чи списку квітів Color List.

Інструментарій FreeHand для малювання й роботи з текстом відповідає необхідним вимогам, але трохи обмежений. В інтерфейсі FreeHand віддано перевагу редагуванню вузлів, а не редагування об'єкта в цілому. Кожна з операцій масштабування, повороту, дзеркального відображення й деформації - виконувані в CorelDraw маніпуляціями в робочому вікні об'єкта - вимагає окремого інструменту з набору інструментарію FreeHand. При виборі об'єкта його точки (вузли) завжди доступні для безпосереднього редагування, але це означає, що ви бачите вузли й траєкторії об'єкта, а не його "закінчений" вид. Ми визнали можливості редагування вузлів цілком застосовними, але типи вузлів були обмежені кутами, кривими й сполучними елементами.

Ми знайшли, що засіб FreeHand для роботи з текстом абзаців надають досить широкі можливості. Ми могли точно вказати, як "обернути" текст навколо гітари, задавши відступи зліва, справа, зверху й знизу. Нам сподобалося, як текст розташовувався навколо об'єктів. Якщо CorelDraw дивним чином розбивав слова, переносячи пару букв nt в слові instrument на новий рядок, навіть, не вставляючи знака перенесення, то FreeHand не робив нічого подібного. FreeHand дозволяє управляти заливанням тексту всередині об'єкта. Ми отримали можливість ретельно підбирати інтервали між буквами, відстані й управляти ними розрядкою тексту, а також компонувати копії, але ширина колонок повинна бути однаковою.

Однак можливості виділення тексту в FreeHand обмежені. Серед усіх програм нашого огляду тільки FreeHand не передбачає градієнтного зафарбовування тексту без попереднього перетворення його в криві. У програмі немає засобів для попереднього перегляду шрифтів, і, хоча ви можете використовувати такі спеціальні ефекти, як зміна масштабу тривимірного тексту, ці ефекти не експортуються у вигляді об'єктів Shockwave. При розміщенні тексту вздовж кривої витрачаються великі зусилля на послідовну, літера за літерою, регулювання між буквених відстаней; ці зусилля виявляються марними, якщо виникає необхідність редагування тексту.

У програмі FreeHand відсутні інструментальні засоби, спеціально призначені для роботи з технічними ілюстраціями. Ми не змогли встановити масштаб і не могли бачити розміри об'єкта при його створенні. Крім того, немає інструментів креслення розмірних чи паралельних ліній.

Засоби виведення для Web, у тому числі експорту файлів GIF і JPEG, працювали бездоганно, проте ми не змогли призначити змішання квітів або визначити палітру. За допомогою невеликої програми, що визначає положення лінійки інструментарію й розмір вікна, можна вставити стиснений формат FHC програми Macromedia Shockwave Graphics в сторінку HTML. Призначаються об'єктів FreeHand уніфіковані покажчики ресурсів URL забезпечують візуальні гіперзв'язки без навігаційних карт. При роботі з Web ви отримуєте доступ до засобів управлінні клацанням правої кнопки миші, а лінійка инструментарію Shockwave містить інструмент, який має вигляд руки, для переміщення зображення "лупу" для збільшення або зменшення його деталей. Afterburner Xtra здійснює стиснення файлів: для наших тестів ступінь стискування становить від 30 до 75%.

FreeHand має дуже продуманим, якщо не найкращим інструментарієм серед усіх представлених пакетів. Але головне достоїнство FreeHand - абсолютне дотримання режиму повній відповідності при виведення зображення (WYSIWYG), без будь-яких неприємних сюрпризів.

Працювати з CorelXara - усе одне, що сидіти за кермом елегантного червоного "Феррарі" із відкидним верхом у красивому весняному парку. Простий і ясний інтерфейс CorelXara перш за все викличе у вас питання: чому вважається, що користуватися пакетами ілюстративної графіки дуже складно?

CorelXara 1.5 - одна з розглянутих у даному огляді програм нового покоління. Вона служить, у першу чергу, для створення графічного зображення на сторінці за один раз і формування блоку тексту за один раз. Програма дозволяє виконувати з малюнками, градієнтним заповненням, зображеннями й діапозитивами такі дії, про які ви могли тільки мріяти. Хоча Corel рекламує CorelXara 1.5 як доповнення до CorelDraw 7 для створення графіки Web, по суті завдяки високій продуктивності, засобів для роботи з Web і спеціалізованому інструментарію CorelXara перевершує CorelDraw у багатьох відношеннях.

Завдяки можливостям масштабування векторної графіки й текстурам растрових зображень, двовимірні об'єкти починають усе більше нагадувати тривимірні. Намалюйте об'єкт. Накладіть текстуру (растрове зображення) або зафарбуйте його (матеріал). Визначте рівень прозорості. Потім перемістіть зображення й відредагуйте на свій смак. Зображення гітари було отримано за допомогою засобу малювання кривих Безьє програми CorelXara. Для побудови кривих ми користувалися "перетягуванням", а для переходу з режиму редагування точок у режим редагування на рівні об'єктів - піктограмами. Було однаково просто змінювати форму, переміщати зображення та об'єднувати біти.

Інтерфейс CorelXara елегантний і простий. Піктограми у верхньому ряду забезпечують доступ до повно кольоровим візуальним набором кольорів, заповнень, штриховок, растрових зображень, шрифтів і графічних вставок (кліпартів). Коли ми вибрали інструмент Transparency на лінійці інструментарію Selector Toolbar зліва, на інформаційної лінійці Infobar згори ми маємо всі необхідні нам засоби управління. Для зображення нашої гітари ми вибрали управління прозорістю щодо чотирьох точок і встановили для кожного кута різні значення. Така градієнтна модель прозорості була накладена на об'єднану групу із двох складних гітар, кожна з яких складалася із трьох об'єктів, двох градієнтних заповнень і растрового зображення - і все це поверх прямокутника фіолетового кольору з розмитими краями, який плавно переходить у блідо-ліловий, накладений на абсолютна прозорість розмитими краями, що в кутку, де вміщено зображення дівчини на пляжі. Здається, що неможливо побудувати настільки складний за описом зображення, проте за допомогою CorelXara це виявилося неважко.

CorelXara полегшує управління кольором шляхом створення сімейств пов'язаних відтінків. Змініть основний колір із синього на зелений, і ваш об'єкт змінить усю гаму відтінків. Зауважте, що CorelXara зовсім позбавлений спеціальних інструментальних засобів для технічних ілюстрацій і, крім того, ви повинні самостійно вводити текст, оскільки в CorelXara не передбачені фільтри імпорту для програм обробки тексту. Проте ця програма була єдиною з розглянутих в огляді, яка дозволяла розміщувати кілька рядків тексту вздовж однієї криволінійної спрямовуючої, а її колекція шрифтів не тільки містить їхні назви, а й показує гарнітури. Програма дуже добре впоралася з передачею кольорового растрового зображення грифа гітари, надавши нам можливість установити параметри спотворень кольору так, щоб на відміну від CorelDraw не створювати новий об'єкт для кожного варіанта кольору.

Найпотужніший на сьогодні інструментальний засіб для графіки Web - зовнішній модуль CorelXara для Netscape Navigator і Microsoft Internet Explorer - дозволяє безпосередньо із браузера збільшувати масштаб зображення до 25 000%. Модуль Shockwave пакету FreeHand фірми Macromedia може виконати таку ж операцію, але CorelXara миттєво виводить змінене зображення на екран навіть на нашому випробувальному ПК, побудованому на базі 133 МГц Pentium. Ви можете призначати об'єктам адреси Web, і вони будуть вести себе подібно зв'язкам, котрий із зовнішнього модуля. Без растрових зображень реклама на розвороті посіла 71 Кбайт (порівняйте з 45-Кбайт файлом JPEG і 154-Кбайт файлом GIF). Завдяки компактності файлу й високої продуктивності перед векторною графікою відкриваються блискучі перспективи в області розробки сторінок Web.

У комплект поставки CorelXara входять шаблони Web і палітри кольорів; тільки за допомогою цього графічного пакета можна безпосередньо створювати анімаційні файли GIF. Ми мали анімацію шляхом багатократних змін малюнка за допомогою засобів Arrange і Create Bitmap Copy і наступного послідовного розташування растрових зображень в "галереї". Анімаційні файли GIF не вимагають наявності зовнішнього програмного модуля й автоматично відтворюються в браузері.

CorelXara може далеко не всі, але в деяких відносинах ця програма не має собі рівних. Якщо ви готуєте складні оригинал-макети, якщо тільки починаєте користуватися пакетами для малювання чи любите працювати із прозорими шарами, CorelXara стане хорошим доповненням до вашого комплекту інструментів.

Намагаючись об'єднати можливості малювання, створення растрових зображень, редагування й верстки сторінок у одній програмі, фірма Deneba Systems розробила пакет Canvas 5, який, реалізуючи численні функції, жодну з них не може виконати бездоганно.

Canvas, попри всіх честолюбних задумах його розробників, не може вважатися повноцінним пакетом ілюстративної графіки. І хоча стверджується, що Canvas об'єднує в собі безліч можливостей, насправді вам уже на самому початку потрібно вибрати тип документа, який ви збираєтеся створювати.

Залежно від обраного типу - Presentation (презентація), Publication (публікація) або Illustration (ілюстрація) - у вас будуть різні можливості й обмеження, а Canvas не дозволяє легко переходити від одного із цих форматів до іншого. Документ типу Illustration може розташовуватися тільки на одній сторінці, але мати кілька шарів (правда, Deneba стверджує, що вас цілком задовольняє одна сторінка для ілюстрації). Документ Publication може займати кілька сторінок, але мати тільки один шар. І якщо ви перейдете з режиму Publication в режим Illustration, то отримаєте тільки першу сторінку документа Publication.

Для роботи в Canvas ми вибрали режим документа Illustration, оскільки нам потрібно управляти шарами, і ми могли при цьому працювати з одною великою сторінкою для побудови своєї тестової електронної таблиці. Canvas виявилася єдиною програмою, яка змогла імпортувати файл LOGO.EPS; вона імпортувала файл Adobe Illustrator з окремими об'єктами для контурів і заповнення.

У той же час намалювати все ту ж гітару для нас було не важко. Інструменти малювання кривих Безьє в Canvas дозволили нам передати форму гітари, переміщаючи криві за допомогою миші. Після цього одним клацанням миші ми змінювали розмір зображення, подвійним клацанням миші переходили в режими обертання й нахилу, а потрійний щиголь миші включав режим редагування, при якому права кнопка миші дозволяє виводити контекстно-залежне меню.

При підготовці попереднього огляду програм ілюстративної графіки у нас склалося враження, що в усіх програм продуктивність залишає бажати кращого. На цей раз тільки Canvas змусив нас чекати. Вивід на екран у великому масштабі зображення кола із градієнтного зафарбовування, оточеного текстом із градієнтним заповненням, обіймав 15 с і більше, причому ця процедура не допускала переривання. Помилки в програмах – це ще одна проблема, з якою ми зіткнулися. Вони раз у раз призводили до повних відмов, але частіше спотворювали зображення на екрані. Наприклад, коли ми встановили режим каркасного зображення, ми могли бачити текст, розташований уздовж криволінійної спрямовуючої, але наші гітари повністю зникли з екрану.

Нам завжди подобалися інструментальні засоби Canvas для технічних ілюстрацій, і нові версії пакета стали винятком. Ми встановили розміри нашого документа по лінійках діалогового вікна щодо "1 дюйм = 1/4 дюйма". Повторне використання фрагментів зображення гітари з допомогою засобу копіювання масивів, екструзії й SmartMouse не викликало труднощів. Надаються широкі можливості простановки розмірних ліній, а якщо ви групуєте його з об'єктами, то вони змінюються належним чином при зміні масштабу.

Перший, і поки єдиний, пакет ілюстративної графіки, що дозволяє редагувати растрові зображення на рівні пікселів, Canvas тепер поповнений новими потужними інструментами створення й редагування растрових зображень. Хоча такі засоби кілька років тому могли вважатися чудовими, сьогодні вони не витримують порівняння з аналогічними засобами в Corel Photo-Paint, Macromedia xRes і Micrografx Picture Publisher, які все сумісні з технологією OLE і входять у якості складових компонентів у більші графічні комплекси, що містять також програми малювання. Тепер в Canvas передбачені фільтри зображень й інструменти для настройки колірних каналів, але на відміну від FreeHand, відсутні зовнішні модулі Photoshop (хоча він і дозволяє працювати з деякими з них) або спеціальні ефекти, як в CorelDraw.

Canvas немає зовнішнього модуля для Web-браузерів, але може експортувати як GIF-, так і JPEG-файли. Однак при виконанні наших тестів Canvas втратив одне із двох вікон із текстом, а в растрових файлах кольору змішали, хоча ми й не призначали змішання квітів. Для робіт з інтенсивною графікою, а також для виведення на друк і на сторінки Web (Deneba стверджує, що незабаром вона запропонує три безкоштовних зовнішніх модуля, які можна буде отримати з її вузла Web), ми рекомендуємо звернути увагу на інші продукти, розглянуті в цьому огляді . Але якщо основна задача полягає в підготовці технічних ілюстрацій, то Canvas може виявитися дуже корисною програмою, яка надає можливості поліпшити оформлення подібних матеріалів.

Революційний підхід до векторної графіки, втілений у пакеті Fractal Design Expression, навів нашій захоплення. Програма надає масу нових можливостей, дозволяючи, зокрема, штрихуванням, зображати вид природних матеріалів. Але оскільки Expression не вважається повно функціональним пакетом, то його краще використовувати в якості доповнення до інших графічними програмами.

Якщо такі програми, як CorelXara і Designer, надають можливість градієнтного зафарбовування вздовж лінії, то Expression відкриває перед вами нові горизонти, дозволяючи розглядати лінію чи контур який спрямовує розміщення нового зображення. Якщо говорити в термінах Expression, у його основу покладена концепція Skeleton Stroke, що становить лінію каркасу (skeleton) і будь-яке графічне зображення як "штрих" (stroke). Уявіть жилки листа у вигляді лінії / каркаса, а форму листа і його розфарбування - як графічного зображення / штриха. Малювання листя стає простою справою та зводиться до малювання ліній.

У ході тестування ми спочатку скористалися вдало реалізованими в Expression інструментальними засобами малювання, щоб отримати зображення гітари. Обриси гітари ми передали за допомогою градієнтного багатоколірного зафарбовування, а для зображення внутрішньої її частини застосували золотаве градієнтне зафарбовування. Інструментальний засіб визначення штриха дав змогу перетворити зображення гітари в графічний елемент (Graphical Element) Skeleton Stroke. Вибравши його з візуального набору Stroke Warehouse і провівши лінію, ми намалювали "скелет" гітари. Викривляючи лінію, ми викривляли гітару. Малюючи прямокутник, розтягували зображення гітари всередині його контуру.

Створюючи точки прив'язки для базової частини й грифа гітари, ми могли витягнути її, не позбавляючи при цьому пізнаваності. Скориставшись іншими прийомами, ми перетворили гітару в повторюваний елемент, а коли намалювали окружність, то на екрані з'явилися три ідеально вигнуті вздовж цього контуру гітари. Ми створили також кілька "ключових" кадрів анімації, переміщаючи точки, поки гітара стала нагадувати жива істота. Після цього Expression сформував "проміжні" кадри, кожним з яких можна користуватися як "штрихом". Ви можете вивести цих кадрів Multi-View Strokes як анімацію як файл AVI або у вигляді послідовно пронумерованих растрових зображень.

Оскільки Expression не містить засобів екструзії ( "видавлювання"), ми створили три гітари для нашого розвороту, користуючись деякими вхідними в комплект "мазками" Natural Media Strokes, що нагадують мазки акварелі, олійної фарби або туші. Кожному з них може бути призначений ступінь прозорості, однак Expression не дозволяє працювати з растровими файлами як об'єктами або зафарбовуваннями.

На відміну від усіх інших програм цього огляду Expression працює тільки з художньо оформленим, а не зі звичайним, що складається з абзаців текстом. Нам довелося вводити свій текст прямо в Expression, і при цьому ми змогли встановити тільки шрифт, вирівнювання й інтерліньяж. Оскільки можливість зменшення між буквеної відстані не передбачена, ми були раді, що текст розташовувався вздовж криволінійної спрямовуючої гарно, й літери на накладалися один на одного. Коли ми імпортували файл Adobe Illustrator, то отримали графіку без тексту. Expression не імпортує файли EPS, подібні файлу нашого логотипу SoundBoard. Крім того, Expression не дозволяє встановлювати масштаб чи креслити розмірні лінії. Не можна визначити справжні розміри об'єкта або відстань між об'єктами. У Expression просто не передбачені засоби створення технічних ілюстрацій.

Управління кольором в Expression обмежене можливостями калібрування його системи, узгодження кольорів по відношенню до системи, застосованої в Adobe Illustrator і старіших версіях Photoshop та CorelDraw. Хоча друкувати зображення можна безпосередньо з Expression, фірма Fractal Design рекомендує створити растровий файл вашого зображення й перенести його в один із зазначених пакетів для друку на принтері, якщо ви реалізуєте кольороподіл CMYK. Ми виконували друк безпосередньо з Expression, дозволивши йому перетворити наше зображення в растрову форму з дозволом 200 крапок / дюйм (щоб зберегти прозорість) перед пересилкою на принтер. Для отримання правильного чорного кольору Expression знадобилося кілька спроб.

Мізерні засоби для безпосереднього виведення на сторінки Web. Нам хотілося б отримати анімаційні файли GIF з Multi-View Strokes пакету Expression, але вдалося створити тільки послідовні файли JPEG. Дійсно, Expression немає засобу експорту GIF, але чудово впорався з файлом JPEG, який ми успішно експортували.

Expression приваблює, головним чином, своєю "революційністю". Імовірно, він не зможе задовольнити всіх ваших потреб у засобах векторної графіки, але це дуже корисний інструмент. Оскільки Expression дозволяє експортувати чудові растрові зображення з будь-яким рівнем дозволу, а всі інші пакети ілюстративної графіки, розглянуті в цьому огляді, тепер можуть імпортувати растрову графіку, то ця програма буде хорошим доповненням до будь-якого набору інструментів.

Ні для кого не секрет - сьогодні, щоб не загубитися на просторах Internet і привернути до себе увагу користувачів, ніяк не можна обійтися без графічного оформлення Web-сторінок і вузлів. Однак тут на шляху розробників виникає проблема: графічні технології для Web не встигають у своєму розвитку за іншими технологіями, і можливості в цій галузі залишаються досить обмеженими.

Справді, два найбільш популярних у даний час графічних форматів Internet - GIF і JPEG - є вже досить старими. Звичайно, невдалими назвати їх ніяк не можна, адже сам факт такого тривалого їхнього існування (наприклад, версія GIF89a використовується з 1989 р.) - свідчення цьому. Але, з іншого боку, навряд чи можна посперечатися з тим, що можливості даних форматів не відповідають сучасним вимогам в області графіки. Так, формат GIF підтримує тільки 256-бітовий колір, а в разі застосування формату JPEG при великій мірі стиснення істотно знижується якість зображення. Крім того, ще в 1995 р. можливість вільного використання GIF виявилася під питанням, коли компанії Unisys, котра має реалізований у цьому форматі алгоритм стискування LZW, і CompuServe, яка розробила сам формат, зібралися стягувати ліцензійні відрахування з кожної програми, що використовує його.

У ситуації, що склалася, група незалежних розробників Internet прийняла рішення про розробку формату, який відповідав би або, навіть, перевершував за своїми можливостями GIF, але був при цьому простим у створенні та повністю мобільним. Новий формат отримав назву Portable Network Graphics (PNG) і був схвалений консорціумом W3C в 1996 р. У грудні минулого року з'явилася його оновлена ​​версія - PNG 1.1.

На жаль, фрактальна компресія, як і JPEG, має істотний недолік: згідно із цими алгоритмами, для аналізу, зображення перед стисненням розбивається на окремі блоки, що ускладнює його поступове промальовування при завантаженні з Web-сайту.

Найбільш перспективні - растрові формати, засновані на алгоритмах wavelet-стиснення. У цій області ведуть розробки практично всі компанії, які займаються створенням графічних форматів. Найперспективнішим є, безумовно, JPEG 2000. Робота над ним ще не завершена, але заявлені параметри вражають: 256 каналів кольору, що дозволить формату працювати з будь-яким колірним простором і підтримувати безліч альфа-каналів: вбудовування ICC-профілів; необмежене поле для метаданих. Але головна перевага wavelet-технології - потоковість. Wavelet-потік можна перервати в будь-який час, при цьому зображення все одно відтворюється, тільки якість його буде залежати від кількості завантажених даних.

Компанія "AT & T" розробила й власний формат на основі wavelet-компресії - DjVu. Його головна особливість - розпізнавання тексту при компресії містить його зображенння й стиснення окремо від графічного й текстового шару. За твердженням компанії, основним призначенням цього формату і є публікація в Web сканованих документів. На сайті AT & T за адресою djvu.research.att.com можна отримати безкоштовний плагін для перегляду DjVu-файлів, а також цілу бібліотеку, яка опублікована в цьому форматі.

Які би не були хороші вище перелічені формати, всіх їх об'єднує один недолік - растр. Наприклад, реалізовані з їхньою допомогою зображення досить складно модифікувати й, навіть, масштабувати. Крім того, незважаючи на використання різних методів стиснення, вони все-таки мають чималий розмір, а отже, і щодо великий час завантаження, що для Web-графіки є особливо критичним.

Векторна графіка заснована не на зберіганні інформації про кожен піксель, а на командах малювання ліній і заповнення форм. Використовується вона вже досить давно, але на відміну від традиційних замкнутих форматів векторні формати для Web побудовані на базі відкритих стандартів, головним чином, мов маркування, у яких для визначення тегів і інших елементів застосовується звичайний текст, що значно спрощує маніпулювання властивостями зображень. Перевагами векторної графіки на основі мов маркування є також можливості вибору, індексування та пошуку елементів зображення й прив'язки її до інших елементів.

Однак говорити про масове впровадження векторної графіки в Web поки що рано, у першу чергу через відсутність єдиного формату.

Найбільш поширеним у даний момент є формат, розроблений компанією Macromedia, - Flash. Завдяки своїм унікальним можливостям його остання (третя) версія дуже швидко завоювала популярність. Flash 3 підтримує анімацію по кейфреймам, морфинг, прозорі об'єкти, гіперпосилання, вбудовування звукових і відео файлів. Засоби для його створення досить прості в користуванні, добре документовані. Плагіни для перегляду поширюються безкоштовно, а розмір вихідних файлів вкрай малий.

Але всі його переваги, на жаль, блякнуть перед одним єдиним недоліком, який змусив Macromedia відмовитися від подальшої розробки формату. Цей недолік - закритість, адже файл Flash - двійковий. Таким чином, його можна редагувати тільки в спеціальній програмі. Тому останнім часом різними компаніями й організаціями запропоновано цілий ряд мовних форматів, і кожен із них претендує на роль єдиного стандарту. У число таких форматів входять Web Schematics, DrawML, PGML і VML.

Формат PNG підтримує 48-бітові кольорові й 16-бітові чорно-білі зображення й забезпечує швидше їхнє завантаження, ніж формат GIF. Він також включає в себе чимало додаткових можливостей, наприклад, альфа-канали (alpha channel), що дозволяють установлювати рівень прозорості для кожного пікселя, і гамма-корекцію. Механізм стиснення зображення в PNG реалізований на базі фільтрів, що дозволяють оптимізувати дані перед стисненням, і алгоритму LZ77, що застосовується в ZIP-архіваторах.

Коли ми зібралися скористатися створеною художньою ілюстрацією для вузла Web, то виявили ще один недолік. CorelDraw експортує кольори так само, як вони відображаються при відключених засобах кольорокорекції. А тому в наших растрових файлах GIF і JPEG були ж самі неправильні кольору, що й при виведенні на принтер HP. Ми з'ясували, що для отримання гарного зображення на сторінці Web, найкраще вибрати режим супердискретизації при експортуванні растрових файлів. Програма перегляду Corel. CMX працювала болісно повільно, а файли CMX виявилися по розміру більші, ніж файли CDR, із чим не можна примиритися при роботі з Web. Зображення гітари, представлене файлом у форматі Barista, просто не могло бути виведене на екран. Barista - розроблений Corel формат на базі мови Java для відображення документів в Web - являє собою перспективну технологію, але в даний час їм краще користуватися тільки для простих документів.

Незважаючи на потужний інструментарій, CorelDraw грішить окремими недоліками. Широкий набір інструментальних засобів робить CorelDraw виключно зручним для малювання, але неприродний вигляд друкованих сторінок і сторінок Web обмежує можливості застосування цього пакета. Якщо ви хочете отримати від CorelDraw все, на що він здатний, ми радимо почекати наступної версії, регулярно перевіряти Web-вузол фірми Corel на наявність нових редакцій і почати з телефонного дзвінка в службу технічної підтримки, щоб переконатися в правильності налаштування засобу кольорокорекції.

Designer 7 наділений безліччю функцій - таких, наприклад, як ітеративне змішування кольорів, - відсутніх у попередніх версіях, але ми, тим не менш, виявили кілька серйозних недоліків. Коли ми "приховували" об'єкт або робили його видимим, втрачалося форматування тексту, а при розташуванні тексту навколо об'єкта виникали неприпустимі перенесення слів. Прив'язка до напрямних здійснювалася лише в разі зміни розмірів об'єкта, але не при його перетягуванні. Крім того, Designer неправильно імпортував наш логотип EPS і перетворив кольору моделі CMYK у кольори RGB при імпортуванні файлу Adobe Illustrator. У програмі не передбачено також кольороподіл і корекції квітів на екрані.

1.3 Наявні алгоритми трекінгу об'єктів та їхнє застосування

У теперішній час, існує велика кількість алгоритмів із розпізнання об'єкта, які мають певні переваги та недоліки. Ми лише розглянемо всього декілька з них.

Трекінгом називається визначення місцеположення рухомого об'єкту (декількох об'єктів) в часі за допомогою камери. Алгоритм аналізує кадри відео і видає положення рухомих цільових об'єктів щодо кадру.

Основна проблема в трекінг складається в зіставленні положень цільового об'єкта на послідовності кадрів, особливо якщо об'єкт рухається швидко щодо частоти кадрів. Таким чином, системи трекінгу зазвичай використовують модель руху, яка описує як може змінюватися зображення цільового об'єкта при всіляких різних його рухах.

Прикладами таких простих моделей руху є:

1. Трекінг плоских об'єктів, модель руху - 2D перетворення (Афінний перетворення або гомографія) зображення об'єкта (наприклад, вихідного кадру);

2. Коли цільовим є жорсткий 3D об'єкт, модель руху визначає вид залежно від його положення в просторі і орієнтації;

3. Для стиснення відео, ключові кадри (key frames) поділяються на макроблоки (macroblocks). Модель руху являє собою розрив ключових кадрів, де кожен макроблок перетворюється за допомогою вектора руху отриманого з параметрів руху;

4. Зображення деформованого об'єкта може бути покрито сіткою (mesh), рух об'єкта задається положенням вершин цієї сітки.

Основне завдання алгоритму трекінгу - це послідовний аналіз кадрів відео для оцінки параметрів руху. Ці параметри характеризують стан цільового об'єкта.

Будемо вважати, що в поточному кадрі за ознакою руху виділені один або більше сегментів. Для цього можна використовувати віднімання фону, оптичний потік, методи на основі суміші гаусовських розподілів і ін. У результаті, замість справжнього напівтонування отримаємо бінарне зображення, як це показано на (рис. 1.1).

У зв'язку з наявністю шуму й різних атмосферних спотворень, результати виділення містять помилки, які легко можна усунути за допомогою методів математичної морфології. Спочатку слід застосувати морфологічне відкриття, що дозволить прибрати мало розмірні сегменти й точковий шум. Потім можна застосувати морфологічне закриття, щоб відновити форму об'єктів. У результаті отримаємо очищене зображення (рис. 1.2), яке в подальшому піддається розмітці і параметризації. Список знайдених сегментів і їхні параметри є вихідними даними для алгоритму стеження.



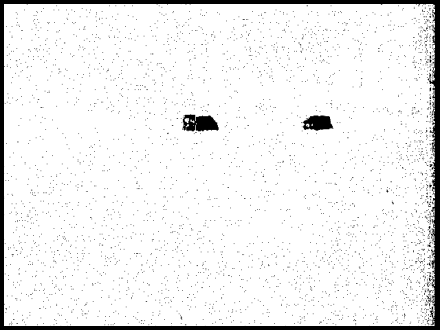


Рис. 1.1 Приклад звичайного зображення та бінарного

Отже, у результаті роботи алгоритму виділення об'єктів може бути отриманий список виявлених на даному кадрі сегментів, проте, щоб вирішити задачу стеження за об'єктами, необхідно зіставити кожному із цих сегментів об'єкт, відомий на попередньому кадрі, або прийняти рішення про виявлення нового об'єкта. Іноді помилково виявлені сегменти є досить стійкими в часі й існують у близьких точках простору протягом декількох кадрів.

З іншого боку, іноді правильно виявлені сегменти короткочасно зникають. Тому повинен бути механізм, який, по-перше, приймає рішення про виявлення нового об'єкта, по-друге, фільтрує нестійкі в часі помилкові сегменти, по-третє, установлює відповідність між відомими раніше об'єктами й новими сегментами, по-четверте, прогнозує координати об'єктів при короткочасному зникненні, по-п'яте, приймає рішення про втрату об'єкта.

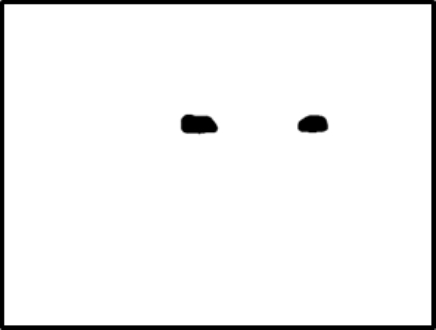


Рис. 1.2 Очищене зображення від шумів

Для додання алгоритму стеження стійкості до тимчасового закриття об'єкта ділянками фону для кожного об'єкта вводиться персональний траєкторний фільтр, головне завдання якого - прогноз координат об'єкту в наступному кадрі на основі аналізу траєкторії руху об'єкта. Зазвичай у якості такого фільтра використовується фільтр Калмана.

При тимчасовому закритті об'єкта ділянками фону або іншими об'єктами уточнення параметрів траєкторії не виконується, і траєкторний фільтр працює в режимі прогнозування координат.

При наявності в послідовності зображень безлічі об'єктів стеження, виникає ймовірність змішування об'єктів у ситуаціях перетину траєкторій їхнього руху. Після перетину траєкторій об'єкти повинні отримувати ті ж ідентифікатори, що були в них до перетину. Приклад правильного призначення ідентифікаторів показаний на (рис. 1.3).





Рис. 1.3 Приклад правильного призначення номерів

В якості першого кроку алгоритму стеження необхідно встановити відповідність між сегментами, знайденими в поточному кадрі, і стеженням за об'єктом. Із цією метою між кожним *i*-м об'єктом і кожним -м сегментом обчислюється кількісна міра подібності. В якості такого заходу можна використовувати евклідову відстань між прогнозованими координатами об'єкта та центром сегменту  :

Слід урахувати, що на зображенні можуть з'явитися нові об'єкти, а простежені протягом деякого часу об'єкти можуть залишати межі кадру або тимчасово затулятися перешкодами. Будемо розглядати три види ситуацій:

а) знайдено відповідність між об'єктом і сегментом;

б) для даного об'єкта не знайдено відповідності серед сегментів;

в) для даного сегмента не знайдено відповідності серед об'єктів.

Евклідову відстань можна розглядати як вартість прийняття рішення (а) про відповідність між i-м об'єктом і j-м сегментом. Введемо величину , як задану вартість рішення (б), і величину , як задану вартість рішення (в).

У підсумку приходимо до вирішення наступного завдання: необхідно встановити відповідність між сегментами й об'єктами або прийняти рішення про неможливість такого зіставлення, таким чином, щоб сумарна вартість усіх рішень була мінімальною. Дане завдання відоме, як задача про призначення, і для її вирішення використовується угорський алгоритм[6].

Щоб застосувати угорський алгоритм, необхідно скласти квадратну матрицю вартості розміру = , де - число відсліткованих об'єктів, а - число знайдених сегментів. Матриця вартостей має наступний вигляд:

E =

де - досить велике число, таке що >> . За рядками матриці відраховуються об'єкти, по стовпцях - знайдені сегменти.

У результаті виконання угорського алгоритму отримаємо список пар

Якщо *t <= і s <=* , то між t об'єктом і s сегментом встановлено відповідність [ситуація (а)].

Якщо *t <= і s >* , то для t об'єкта не знайдено відповідного йому сегмента [ситуація (б)].

Якщо *t > і s <=* , то для s сегмента не знайдено відповідного об'єкта [ситуація (в)].

Кожна з перерахованих ситуацій призводить до необхідності виконання різних дій. У першому випадку слід оновити список параметрів об'єкта на основі параметрів відповідного йому сегмента. У другому випадку оновлення неможливе, однак відсутність відповідного сегменту може пояснюватися тимчасовим закриттям об'єкта. Тому в цьому випадку слід перейти до прогнозування параметрів об'єкта на основі попередніх спостережень. Для уточнення й прогнозування координат об'єктів використовується фільтр Калмана. Нарешті, у третьому випадку сегмент відповідає знову знайденому об'єкту.

У ході спостереження об'єкти можуть залишати межі кадру. У цьому випадку слід припинити стеження за ними. Критерієм визначення подібних ситуацій є неможливість протягом досить тривалого часу встановити відповідність між даним об'єктом і будь-яким із виділених у поточному кадрі сегментів.

Звичайно, описана вище схема досить проста. У більш складних сценаріях потрібно було б обробляти закриття об'єктів шляхом злиття й поділу треків.

Наступний алгоритм, заснований на використанні бібліотеки «OpenCV»[4, с. 30]. Спочатку йде запуск основної програми, через яку буде здійснюватися розпізнавання об’єкта через вебкамеру. Після запуску з’являється вікно, яке має назву «OpenCV». На це вікно транслюється передача зображення з вебкамери користувача. Потім іде перетворення з RGB зображення в HSV для розпізнавання об’єкта. Паралельно із цим виконується робота програмного коду з розпізнаванням об’єкта певного кольору, наприклад, це червоний колір. Якщо на зображенні з вебкамери присутній даний колір, тоді йде його виділення в жовту рамку. Разом із цим виконується відтворення координат об’єкта в основному вікні «OpenCV».

На старті програми, виконується ініціалізація вебкамери, та створюється вікно, у якому виводиться зображення у режимі реального часу.

cvNamedWindow( "OpenCV", 1 );

VideoCapture cap(0);

Після цього, розпочинається робота із самим зображенням, а саме захоплюється зображення з камери, і поміщається в змінну.

cap >> img;

Потім, потрібно виконати конвертацію зображення з RGB у HSV і вивести це зображення на екран. В RGB зображенні всього три кольори: червоний, зелений, та синій. У процесі відображення, кількість змішування цих кольорів визначає значення пікселя. А у HSV зображенні, усього один білий колір, який буде використаний для окремого показу у вікні жовтих пікселей.

cvtColor(img, hsv, CV\_BGR2HSV);

imshow("HSV",hsv);

Тепер отримаємо бінарне зображення, яке виведемо на екран у окремому вікні, і за допомогою бінарного перетворення, задамо діапазон пошуку червоних пікселей. Для підбору потрібного кольору, була використана палітра кольорів у програмі «Paint».

inRange(hsv, Scalar(20, 100, 100), Scalar(30, 255, 255), binary);

imshow("Binary",binary);

Після цього, знаходимо контури бінарного зображення.

int i;

vector< vector<Point> > contours;

findContours(binary, contours, CV\_RETR\_TREE, CV\_CHAIN\_APPROX\_SIMPLE);

vector<double> areas(contours.size());

Коли вже контури знайдені, потрібно визначити площу найбільшого контуру.

for(i = 0; i < contours.size(); i++)

{

areas[i] = contourArea(Mat(contours[i]));

}

Далі отримуємо індексивусіх найбільших контурів, для того, щоб зробити їх відображення.

double max;

Point maxPosition;

minMaxLoc(Mat(areas),0,&max,0,&maxPosition);

// Малюємо великий контур

drawContours(binary, contours, maxPosition.y, Scalar(255), CV\_FILLED);

Після того як уже здійснився пошук контурів бінарного зображення, а також проведена їхня індексація, потрібно виконати виділення жовтих пікселей. Тому для цього, кожен піксель, або увесь об'єкт жовтого кольору, виділяємо червоним прямокутником.

Point center;

Rect r;

if (contours.size() >= 1)

{

r = boundingRect(contours[maxPosition.y]);

rectangle(img, r.tl(),r.br(), CV\_RGB(255, 0, 0), 5, 5, 0); // малюємо прямокутник

}

Знаходимо центр та визначаємо координати об'єкта жовтого кольору, і виводимо їх на робоче вікно «OpenCV», а також на головне вікно програми.

// знаходимо центр об'єкта

center.x = r.x + (r.width/2);

center.y= r.y + (r.height/2);

// відображаємо координати x та y на зображення

char x[15],y[6];

itoa(center.x,x,10);

itoa(center.y,y,10);

strcat(x,",");

putText(img, strcat(x,y), center, FONT\_HERSHEY\_COMPLEX\_SMALL, 0.8, cvScalar(255,0,0), 1, CV\_AA);

imshow("OpenCV",img);

printf("\n Point x,y= ");

printf(strcat(x,y));

**2**

**1**

8. Мала площа контуру з

бінарного зображення

7. Пошук великої

площі контуру

6. Знаходження червоних

пікселей

5. Конвертування з HSV

зображення в бінарне

4. Конвертування з RGB

зображення в HSV

2. Ініціалізація

камери

Початок програми

1. Створення вікна

«OpenCV»

3. Захоплення зображення з вебкамери

9. Малювання великого контуру жовтими пікселями з бінарного зображення

10. Розмір

контуру більше

12. Отримання центра

зображення

13. Виведення координат x і y yj,hf;tyy

Кінець програми



**2**

11. Малювання прямокутника червоним

кольором

**1**

Рис. 1.4 Алгоритм розпізнавання об`єкта

## 1.4 Висновки до розділу 1

1. Проведено аналіз існуючих засобів створення векторної графіки. Розглянуті відомі методи та програми, і було виконано аналіз їхніх переваг і недоліків.

2. Один з обраних алгоритмів більш складний, але більше легкий у програмуванні та дає більшу швидкість викання процедур і тому він більш прийнятний для досягнення поставленої мети.

# РОЗДІЛ 2 ПРОЕКТУВАННЯ ВІДЕОПЕРА

## 2.1 Алгоритм програмного забезпечення роботи відеопера

В даній дипломній роботі, була повністю реалізована поставлена задача, шляхом розробки програмного забезпечення відеопера на основі створеного автором алгоритму (рис. 2.1).

Початок програми

1.Ініціалізація камери

2. Захват зображення з камери

3. Конвертування з RGB

зображення в HSV

4. Підключення палітри для малювання

5. Дзеркально відображаємо зображення

6. Знаходження жовтих пікселей

6. Фільтрація зображення від шумів

7. Очікування дій від користувача

Взаємодія користувача з палітрою

"ESC"

8. Малювання лінії

Кінець програми

Рис. 2.1 Алгоритм програми "відеоперо"

Спочатку іде ініціалізація камери, де виконується перевірка на її готовність.

CvCapture\* capture = 0;

capture = cvCaptureFromCAM(0);

if(!capture)

{

cout << "Could not initialize capturing...\n";

return -1;

}

Тобто, якщо камера готова до використання, а саме підключена до комп'ютеру, і встановлені потрібні драйвери, то далі виконується програмний код, а в іншому випадку програма закінчить свою роботу.

IplImage\* GetImage(IplImage\* img, CvScalar& lowerBound, CvScalar& upperBound)

{

// Конвертация входящего изображение в HSV

IplImage\* imgHSV = cvCreateImage(cvGetSize(img), 8, 3);

cvCvtColor(img, imgHSV, CV\_BGR2HSV);

IplImage\* imgThreshed = cvCreateImage(cvGetSize(img), 8, 1);

cvInRangeS(imgHSV, lowerBound, upperBound, imgThreshed);

cvReleaseImage(&imgHSV);

return imgThreshed;

}

Після вдалої ініціалізації камери, виконується конвертування зображення з "RGB" у "HSV" для розпізнавання об'єкту.

Далі задається діапазон кольору об'єкту для подальшого його розпізнавання.

CvScalar lowerBound = cvScalar(20, 100, 100); // yellow

CvScalar upperBound = cvScalar(30, 255, 255);

Потім іде підключення палітри для малювання (рис. 2.2). А саме кольори ліній, їхня товщина й також інструменти для взаємодії із самою програмою.

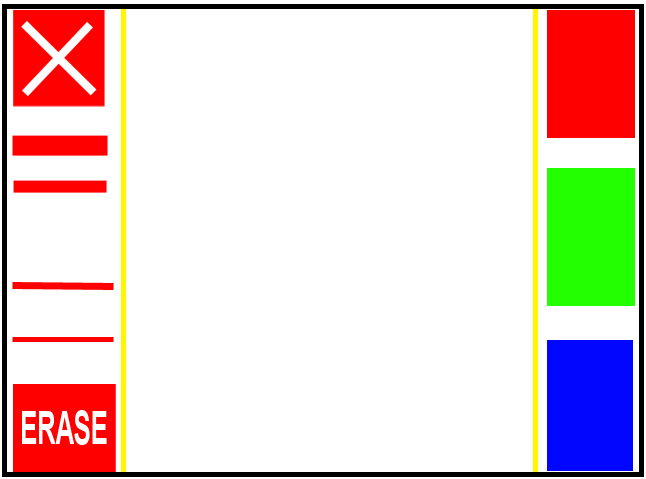


Рис. 2.2 Інструментарій користувача

Після цього, застосовується фільтрація зображення від шумів, які можуть з'явитися від веб-камери за умови, що саме обладнання є низької якості.

cvSmooth( frame, frame,

CV\_MEDIAN,

5, 5 //размер фильтра

);

Далі програма очікує дій від користувача. Якщо користувач обере закриття програми, тобто наведе у верхній лівий кут (рис. 2.2) або натисне на клавіатурі клавішу "ESC", то програма закриється.

else if(posX > 0 && posX < 90 && posY > 0 && posY < 120) // выход

{

sprintf (buffer, "EXITING in %d",confirm\_close);

cvPutText( frame, buffer, cvpoint, &font, red );

confirm\_close--;

if(confirm\_close < 0)

break;

}

Коли ж користувач обере колір, товщину лінії, то "відеоперо" прийме ці характеристики, а користувач побаче на екрані результат цих дій.

else if(posX > 540 && posY > 360) // синий

{

lineColor = blue;

cvPutText( frame, "Blue color selected.", cvpoint, &font, blue );

}

else if(posX > 540 && posY > 200 && posY < 280) // зеленый

{

lineColor = green;

cvPutText( frame, "Green color selected.", cvpoint, &font, green );

}

else if(posX > 540 && posY < 120) // красный

{

lineColor = red;

cvPutText( frame, "Red color selected.", cvpoint, &font, red );

}

else if(posX < 90 && posY > 130 && posY < 390) // толщина линии

{

lineThickness = 6 - ( posY/60-1 ); // меняем толщину линии на основе posY

}

sprintf (buffer, "%d",lineThickness);

cvPutText( frame, buffer, cvPoint(40,255), &fontbig, lineColor );

double diff\_X = lastX-posX;

double diff\_Y = lastY-posY;

double magnitude = sqrt( pow(diff\_X,2) + pow(diff\_Y,2) );

if(magnitude > 0 && magnitude < 100 && posX > 120 && posX<530)

{

// рисование линии при условии что она находится в рабочей зоне

cvLine(imgDrawing, cvPoint(posX, posY), cvPoint(lastX, lastY), lineColor, lineThickness,CV\_AA);

}

Також, користувачеві доступна функція ластику, тобто затирання ліній. Ця функція працює за принципом роботи в програмі "Paint". Коли користувач обере функцію повної очистки дошки від намальованих ліній, то через 10 секунд, дошка буде повністю очищена. Така затримка в часі потрібна задля того, щоб виключити помилкове обрання.

if(posX < 90 && posY > 400) // очищаем

{

lineColor = white; // очищение доски белым цветом

cvPutText( frame, "Eraser selected.", cvpoint, &font, white );

sprintf (buffer, "Clearing the screen in %d",confirm\_clear); // обратный отсчет для полной очистки доски

cvPutText( frame, buffer, cvPoint(150,70), &font, red );

confirm\_clear--;

if(confirm\_clear < 0)

{

confirm\_clear = 20;

sprintf (buffer, "d0%d.jpg",image\_num++);

cvSaveImage(buffer ,imgDrawing);

ScreenClear(imgScribble,imgDrawing);

cvPutText( frame, "Cleared the screen.", cvPoint(150,110), &font, white );

}

}

Окрім цього, програма постійно оновлює та запам'ятовує координати останнього положення "відеопера" для того, щоб коректно відображати інформацію, та вимальовувати лінії по отриманим координатам.

## 2.2 Інструментарій розробки відеопера

Для вирішення поставленої задачі було обрано бібліотеку «OpenCV». Це бібліотека, написана мовою високого рівня (C / C ++), яка містить алгоритми для інтерпретації зображень, калібрування камери за еталоном, усунення оптичних спотворень, визначення подібності, аналіз переміщення об'єкта, визначення форми об'єкта й стеження за об'єктом, 3D-реконструкцію, сегментацію об'єкта, розпізнавання жестів і т.д.

Офіційно проект «OpenCV» був запущений у 1999 році за ініціативою Intel Research із ціллю розвивати [CPU](http://uk.wikipedia.org/wiki/CPU)-ресурсномісткі програми. Основними вкладниками в проект була Intel's Performance Library Team та певна кількість експертів із чисельної оптимізації в Inter Russia. На перших етапах розвитку OpenCV основними задачами бібліотеки були:

* Розвивати дослідження в напрямку комп'ютерного зору, забезпечуючи добре оптимізований та відкритий код бібліотеки.
* Поширювати знання в сфері комп'ютерного зору, забезпечуючи загальну інфраструктуру, яку б могли розвивати розробники, і, таким чином, код ставатиме більш легким для сприйняття та обміну.
* Розвивати засновані на роботі з комп'ютерним зором комерційні програми, створюючи незалежну від платформи, оптимізовану та безкоштовну бібліотеку. Для цього використовувалася ліцензія, яка не вимагала від таких комерційних додатків бути відкритими.

Перша альфа-версія «OpenCV» була оприлюднена на IEEE конференції з комп'ютерного зору й розпізнавання образів у 2000 році, і п'ять бета-версій було випущено в період між 2001 і 2005 роками. Перша версія 1.0 була випущена в 2006 році. У середині 2008 року, «OpenCV» отримала корпоративну підтримку від [Willow Garage](http://uk.wikipedia.org/w/index.php?title=Willow_Garage&action=edit&redlink=1) і знову перейшла у стадію активної розробки. «Предрелізна» версія 1.1 була випущена у жовтні 2008 року.

Другий великий випуск «OpenCV» відбувся в жовтні 2009 року. «OpenCV 2» включала в себе серйозні зміни в інтерфейсі [C++](http://uk.wikipedia.org/wiki/C%2B%2B). Ці зміни спрямовані на більш прості, тип-безпечні моделі, додавання нових функцій, і кращу реалізацію існуючих моделей у плані швидкодії (особливо на багатоядерних системах). Офіційні релізи надалі відбуваються кожні 6 місяців. Іхньою розробкою займається незалежна команда з Росії, яка підтримується комерційними корпораціями.

У серпні 2012 року, підтримку «OpenCV» було передано некомерційній організації, OpenCV.org.

Ця бібліотека популярна за рахунок своєї відкритості й можливості безкоштовно використовуватися як у навчальних, так і комерційних цілях.

Фактично, «OpenCV» - це набір типів даних, функцій і класів для обробки зображень алгоритмами комп'ютерного зору.

Основні модулі бібліотеки:

*cxcore* – ядро, яке містить базові структури даних і алгоритми:

• базові операції над багатовимірними числовими масивами;

• матрична алгебра, математичні функції, генератори випадкових чисел;

• Запис / відновлення структур даних в / з XML;

• базові функції 2D графіки;

*CV* - модуль обробки зображень та комп'ютерного зору:

• базові операції над зображеннями (фільтрація, геометричні перетворення, перетворення колірних просторів і т. д.);

• аналіз зображень (вибір відмітних ознак, морфологія, пошук контурів, гістограми);

• аналіз руху, стеження за об'єктами;

• виявлення об'єктів, зокрема, осіб;

• калібрування камер, елементи відновлення просторової структури;

*Highgui* - модуль для введення / виведення зображень і створення користувальницького інтерфейсу:

• захоплення відео з камер і з відео файлів, читання / запис статичних зображень;

• функції для організації простого UI (всі демо-програми використовують HighGUI);

*Cvaux* - експериментальні й застарілі функції:

• стерео калібрації;

• саме калібрації;

• пошук стерео відповідності;

• знаходження й опис рис обличчя.

*CvCam* - захоплення відео:

• дозволяє здійснювати захоплення відео із цифрових відеокамер (підтримка припинена й в останніх версіях цей модуль відсутній).

Як видно з опису модулів бібліотеки «OpenCV» - вона надає широкі можливості для реалізації поставленого в курсовій роботі завдання, а саме організації відстеження об'єкта на відео.

Бібліотеку «OpenCV» можна завантажити за адресою - <http://sourceforge.net/projects/opencvlibrary/files/opencv-win> (рис. 2.3). В даній роботі використовувалася версія 2.4.9. Нижче приведено приклад.

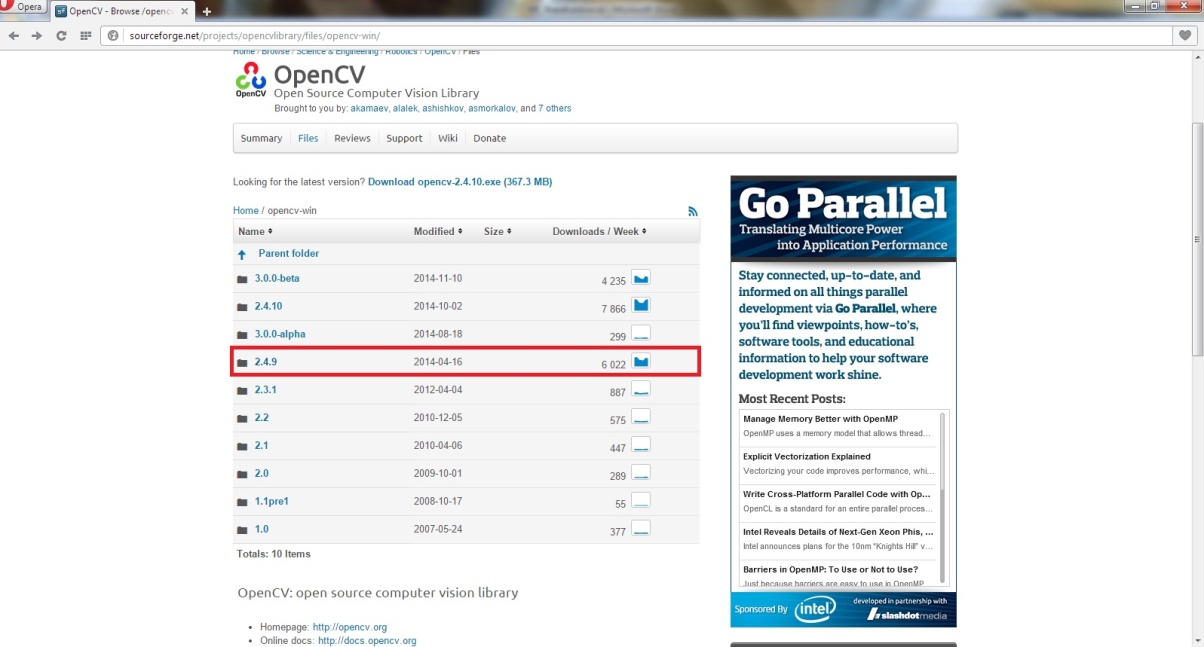


Рис. 2.3 Сайт завантаження бібліотеки «OpenCV»

Після завантаження вказуємо папку, у яку необхідно розпакувати бібліотеку (краще розпакувати бібліотеку просто на диск **С:\**) та натискаємо кнопку **Extract**. Нижче приведений приклад.

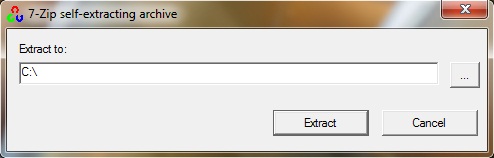


Рис. 2.4 Приклад розпакування бібліотеки «OpenCV»

Далі виконаємо заміну системної змінної **Path**. Для цього нам потрібно записати шлях, який показано нижче на (рис.2.5).

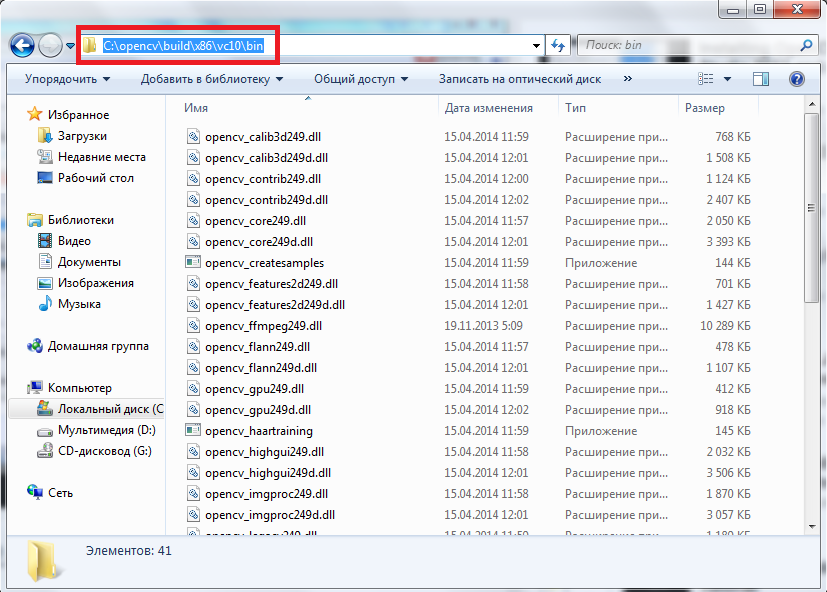


Рис. 2.5 Вікно шляху для налаштування роботи с бібліотекою «OpenCV»

Потім додаємо даний шлях у властивості системи: **Мій комп’ютер, Властивості** системи, вибираємо пункт **Додаткові параметри системи**, натискаємо кнопку **Змінні середовища (**рис.2.6).

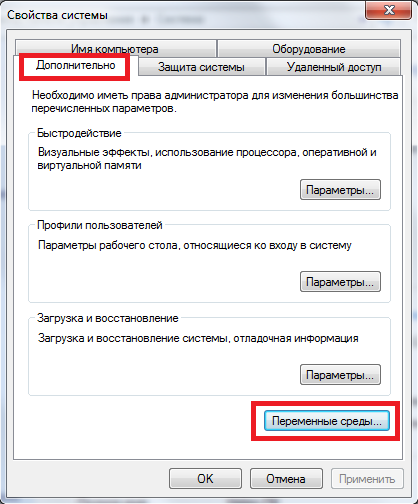


Рис. 2.6 Вікно властивостей системи

Далі вибираємо системну змінну **Path** і натискаємо кнопку **Змінити…**(рис. 2.7).

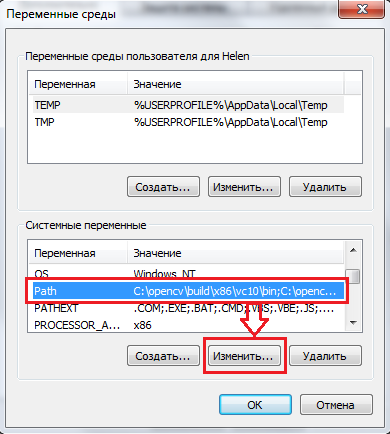


Рис. 2.7 Вікно системних змінних

Тепер потрібно змінити шлях системної змінної **Path** на шлях, де встановлена програма. У нашому випадку, це буде **C:\opencv\build\x86\vc10\bin;** (рис.2.8).

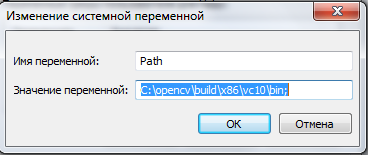


Рис. 2.8 Вікно зміни шляху системної змінної «Path»

Натискаємо **Ок** та робимо перезавантаження комп’ютера. На цьому налаштування бібліотеки «OpenCV» для роботи на комп'ютері, закінчилась. Тепер переходимо до налаштування проекту.

Далі треба створити нове Рішення (Solution), у яке включимо перший Проект (Project). Послідовно виконуємо наступні кроки:

1. Запускаємо програму **«Microsoft Visual Studio 2010»**
2. Створюємо новий проект **(File → New → Project)**.
3. Як показано на (рис. 2.9)., в діалоговому вікні **New Project** в типах проекту виберіть **Win32**, в шаблонах **Win32 Console Application**, в полі **Name** введіть назву програми (для кожної програми буде використано своя назва), в поле **Solution Name** - назва рішення OpenCV, в полі **Location** вказуємо шлях до папки, в яку бажаємо зберегти свій проект. Натискаємо **OK**.

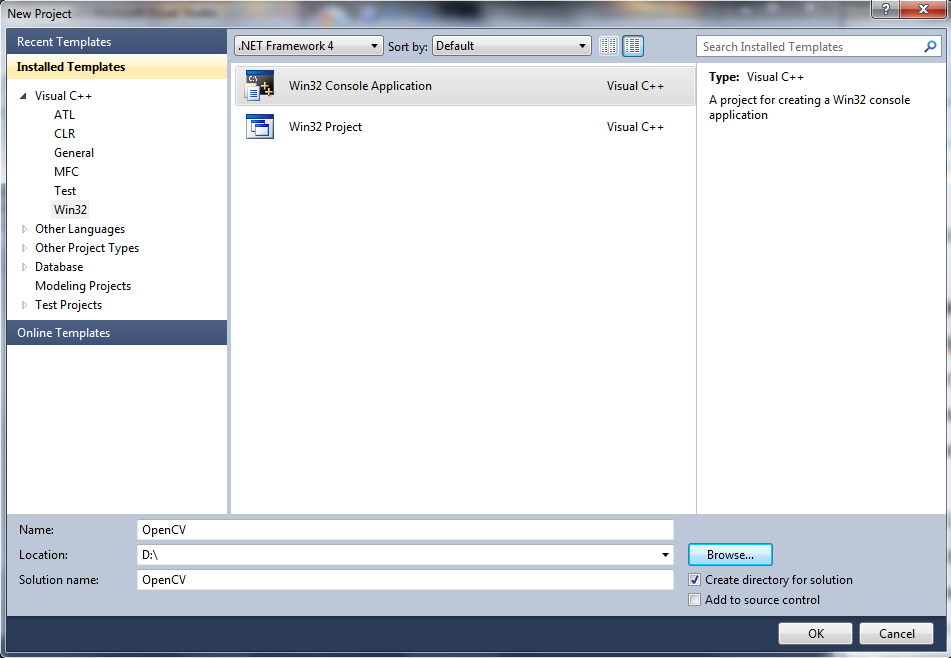


Рис. 2.9 Вікно створення нової програми

1. У діалоговому вікні **Win32 Application Wizard** натискаємо **Next** (або вибираємо **Application Settings** в дереві ліворуч) і встановлюємо прапорець **Empty Project**. Натискаємо **Finish** (рис. 2.10).

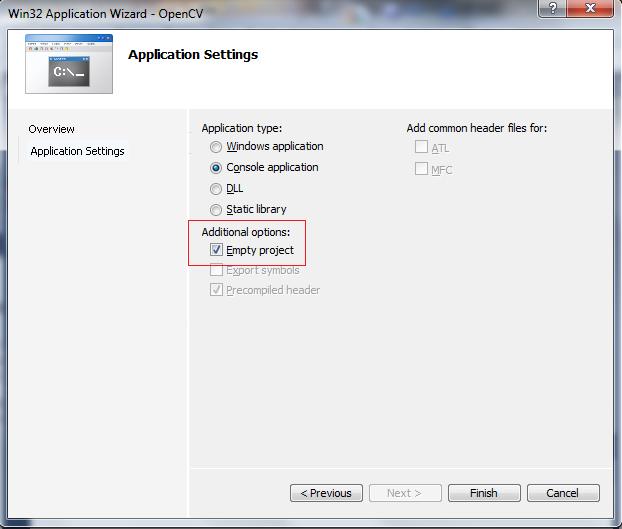


Рис. 2.10 Діалогове вікно «Win32 Application Wizard»

1. У вікні **Solution Explorer** в папці **Source Files** виконуємо команду контекстного меню **Add → New Item ....** У дереві категорій зліва вибираємо **Code**, у шаблонах праворуч **-** **C ++ File (.cpp)**, в полі **Name** вводимо ім'я файлу **opencv**, натискаємо **Add**. У результаті виконаної послідовності дій, у вікні редактора коду Visual Studio, буде відкритий порожній файл **opencv.cpp**. Туди записуємо код програми.

Далі переходимо до налаштування властивостей проекту. Процес налаштування властивостей проекту зводиться до виконання чотирьох дій:

* Виконуємо команду контекстного меню **Properties**, щоб отримати доступ до налаштувань проекту. Відкриваємо вкладку **Configuration Properties → VC++ Directories** (рис. 2.11). У полі **Include Directories** вказуємо шлях **C:\opencv\build\include**. У полі **Library Directories** вказуємо шляхи **C:\opencv\build\x86\vc10\bin; C:\opencv\build\x86\vc10\lib**. Та в полі **Source Directories** вказуємо шлях **C:\opencv\sources\modules\highgui\src**. У нашому випадку **C:\** - це локальний диск, на який була встановлена бібліотека «OpenCV». Натискаємо кнопку **Apply**, щоб застосувати вказану властивість. Результат програми повинна мати приблизно такий вигляд, який показано нижче.

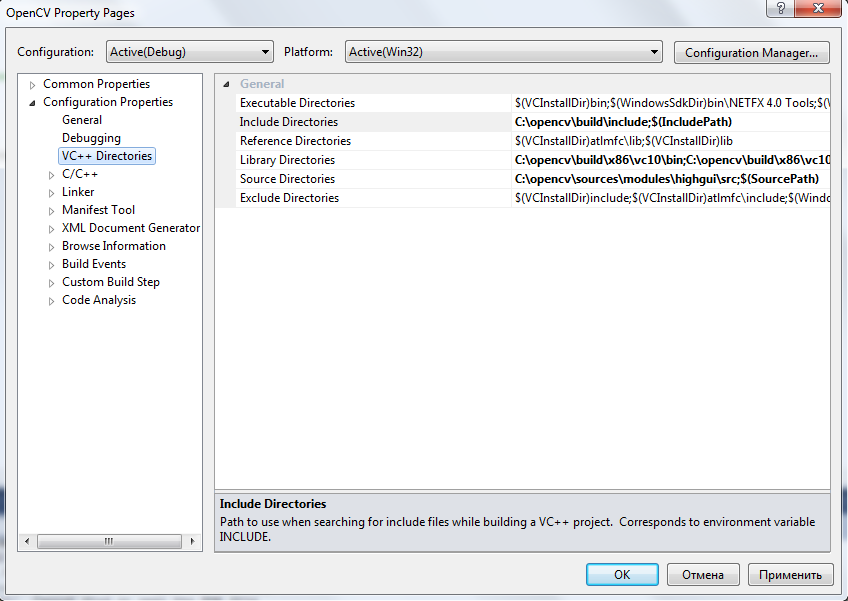


Рис. 2.11 Результат зображення додавання деяких шляхів.

* Установка шляхів до заголовків файлів бібліотеки «OpenCV». Відкриваємо вкладку **Configuration Properties → С/C++ → General** (рис. 2.12). У полі **Additional Include Directories** вказуємо шляхи до заголовків файлів бібліотеки «OpenCV» **C:\opencv\build\include; C:\opencv\build\include\opencv**.

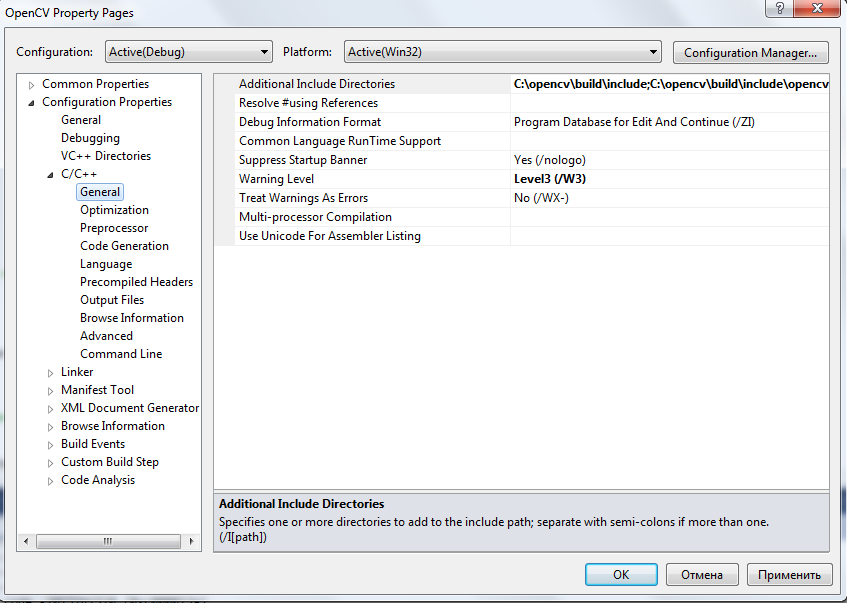


Рис. 2.12 Вікно властивостей проекту для установки шляхів до заголовків файлів підключених бібліотек

* Установка шляхів до підключених бібліотек. Відкриваємо вкладку **Configuration Properties → Linker → General** (рис. 2.11.). У полі **Enable Incremental Lincking** змініть **Yes (/INCREMENTAL)** на **No (/INCREMENTAL:NO)**. У полі **Use Library Dependency Inputs** змініть **No** на **Yes**.

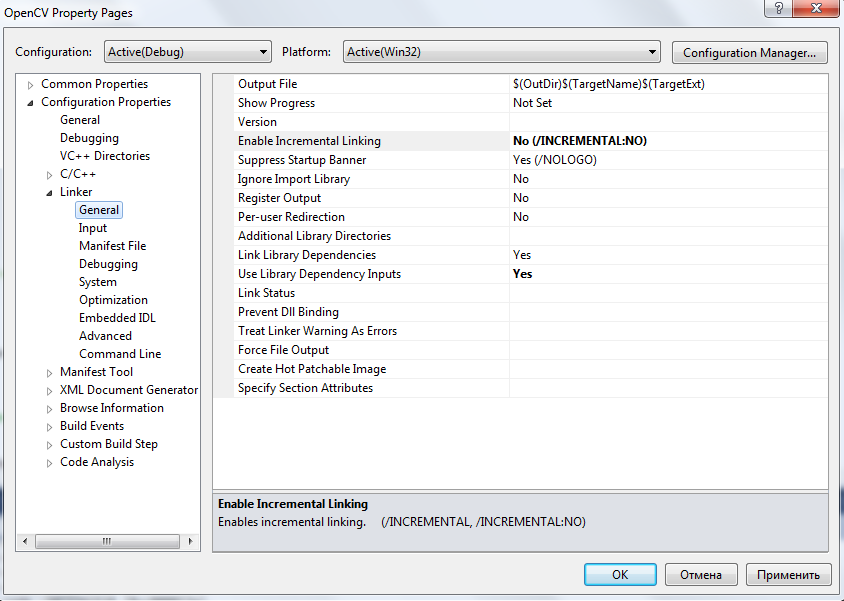


Рис. 2.13 Вікно властивостей проекту для установки шляхів до підключених бібліотек

* Вказівка списку підключаються програмних бібліотек. Відкрийте вкладку **Configuration Properties → Linker → Input** (рис. 2.12.). Схема іменування lib-файлів наступна: **opencv\_ <мод> <вер> d.lib**, де **<мод>** - назва модуля, **<вер>** - версія бібліотеки (наприклад, opencv\_core249d.lib). Відзначимо, що можна підключати не всі модулі бібліотеки, а тільки ті, функціонал яких буде використаний у процесі розробки програми. Установлюємо в полі **Additional Dependencies** список lib-файлів: «opencv\_core249d.lib; opencv\_imgproc249d.lib; opencv\_highgui249d.lib; opencv\_ml249d.lib; opencv\_video249d.lib; opencv\_features2d249d.lib; opencv\_calib3d249d.lib». Натискаємо кнопку **Apply**, щоб застосувати вказану властивість, потім натискаємо клавішу **Ок**.

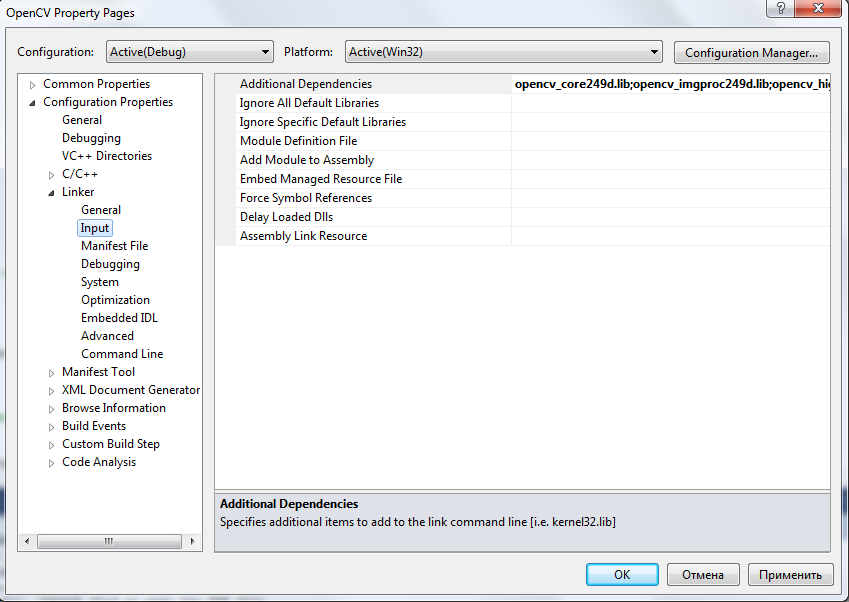


Рис. 2.14 Вікно властивостей проекту для установки списку підключених бібліотек

## 2.3 Висновки до розділу 2

1. Підготовлено середовище розробки ПЗ відеопера.

2. Більш детально розглянуті властивості та суть бібліотеки для комп'ютерного зору "OpenCV".

3. Створено програмний код для реалізації поставленої мети кваліфікаційної роботи.

4. Побудовано алгоритм програмного забезпечення відеопера.

# РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ВИПРОБОВУВАННЯ ВІДЕОПЕРА

## 3.1 Інтерфейс та інструкція для користувача

Для успішного запуску програмного забезпечення відеопера, потрібен комп'ютер із налаштованою до роботи вебкамери, або ноутбук із вбудованою камерою.

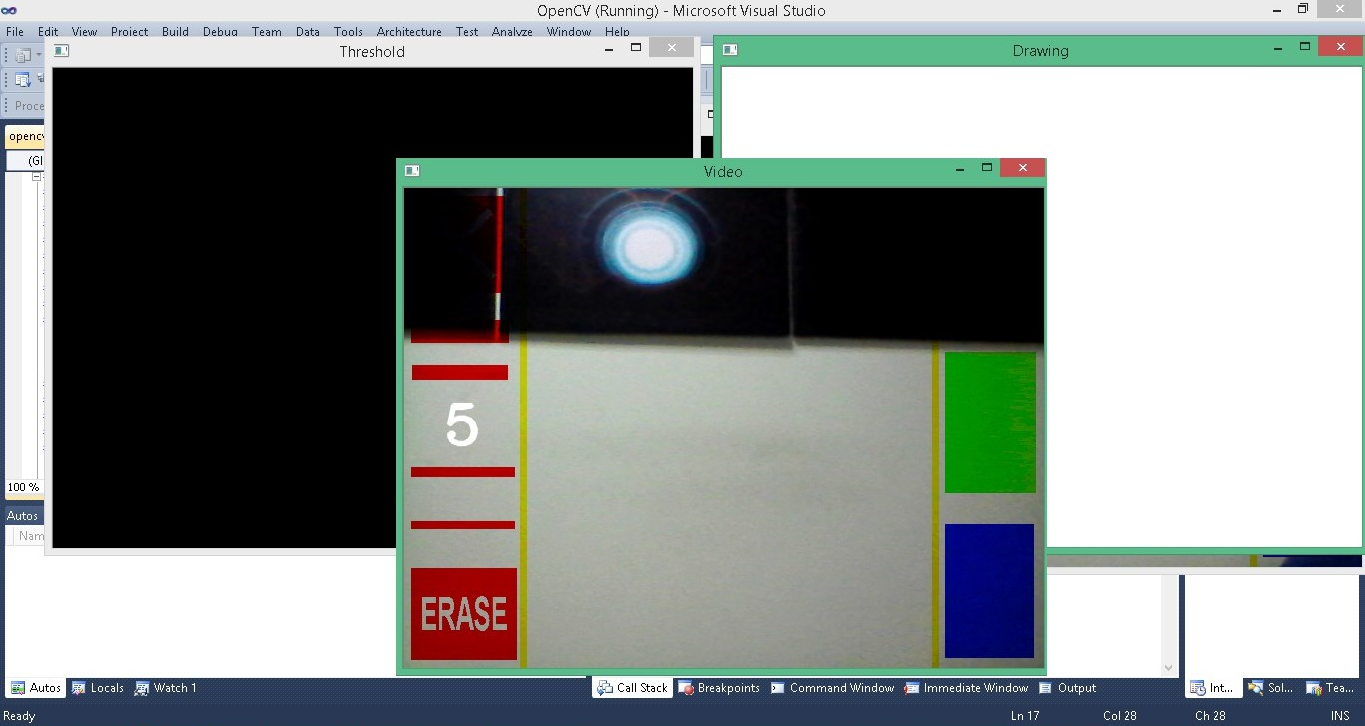
Після старту програми, іде перевірка на правильно налаштоване обладнання, і якщо все вірно налаштовано, тоді з'являються вікна програмного забезпечення (рис. 3.1).

Рис. 3.1 Старт програми

Тепер потрібно взяти "перо" жовтого кольору, та піднести до камери, й у вікні під назвою "Threshold" з'явиться зображення білого кольору, це буде означати, що програма повністю готова до виконання завдань користувача (рис. 3.2).

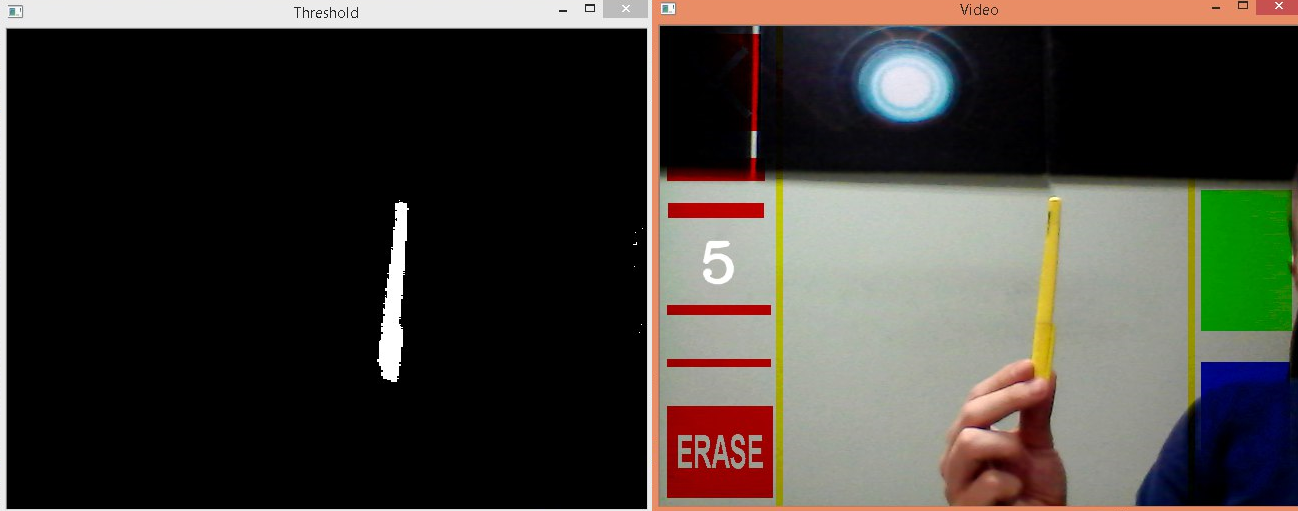


Рис. 3.2 Визначення кольору об'єкта

Перед нами представлений інструментарій користувача, де користувач може обрати властивості майбутньої лінії, а саме її колір, та товщину. Для вибору представлені три кольори: червоний, зелений, синій. (рис. 3.3). Щоб обрати один із кольорів, користувачеві потрібно навести "перо" на відповідні прямокутники, які мають певний колір. При виборі кольору, перед користувачем з'явиться надпис "Color selected", який говорить, що був обраний певний колір.

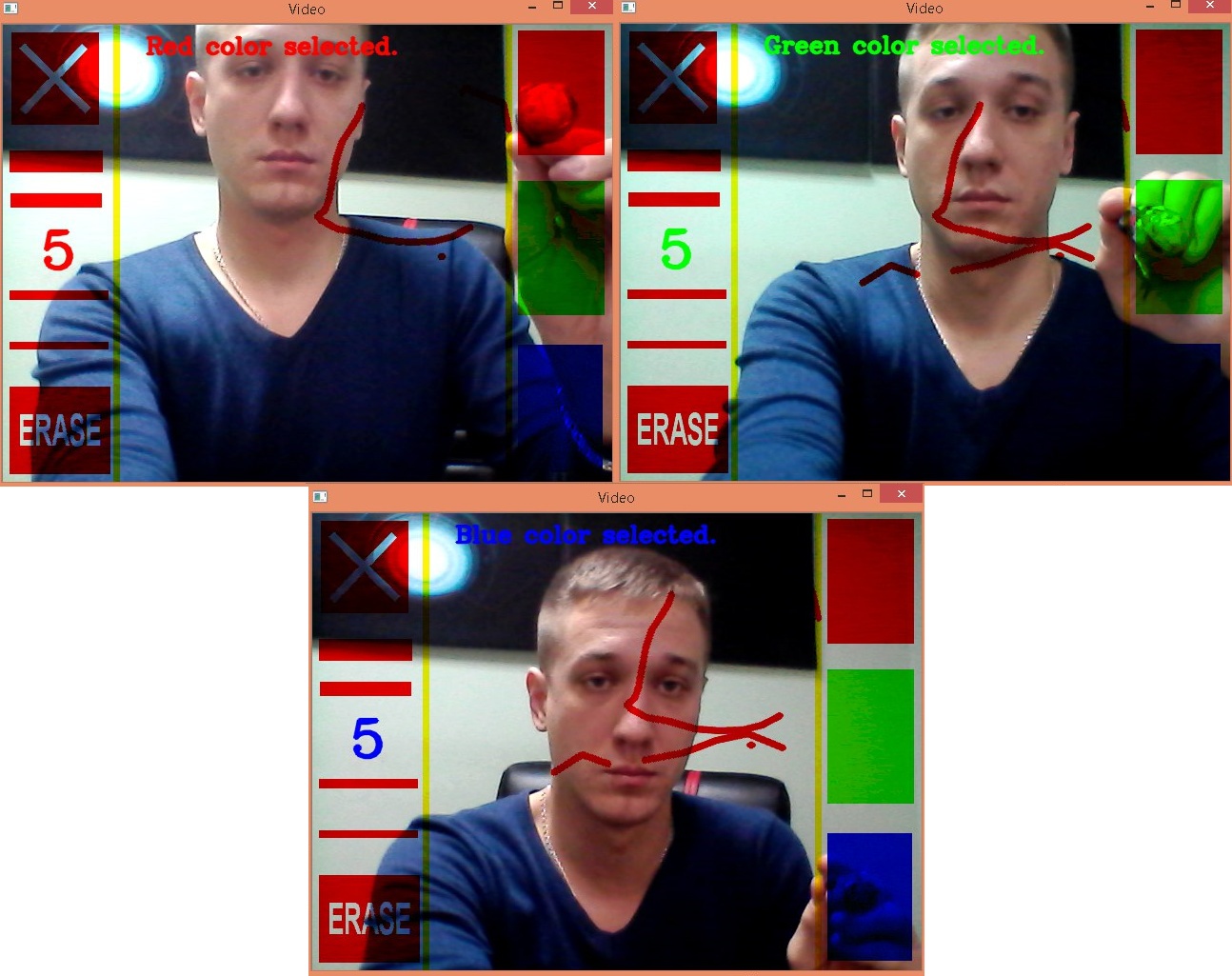


Рис. 3.3 Вибір кольору

Після чого, потрібно обрати товщину лінії, які представлені з лівої сторони панелі користувача (рис. 3.3).

Також у даному програмному забезпечені реалізована функція закриття програми. Щоб закрити програму, потрібно навести "перо" на лівий кут зверху панелі (рис. 3.4). Якщо користувач наведе "перо" у лівий кут, тоді з'явиться відповідний напис "EXITING in ..." та піде відлік часу, і через 10 секунд програма закриється. Також є інший варіант, де потрібно на клавіатурі натиснути клавішу "ESC".

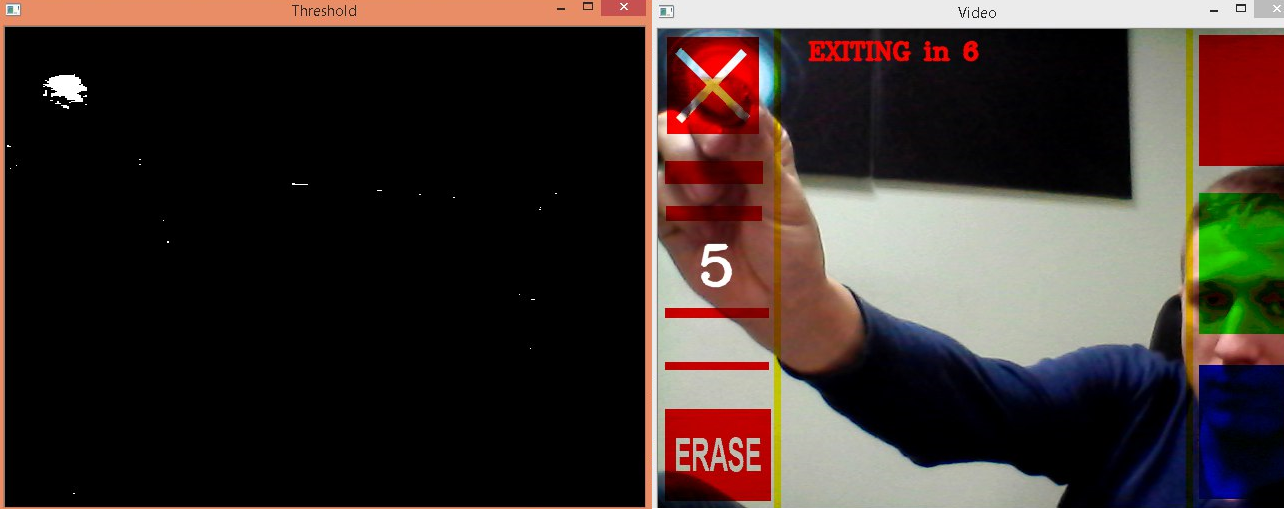


Рис. 3.4 Закриття програми

Понад тим, у даному програмному забезпеченні, реалізована функція "стирання". Тобто, якщо користувач обере дану функцію, то замість "олівця", у користувача з'явиться "гумка", якої він має змогу зітерти з дошки те, що він намалював.

Для активації цієї функції, користувачеві потрібно навести "перо" у нижній лівий кут.

Для показу роботи, спробуємо намалювати приклад додавання "2+2=4" (рис. 3.5), після чого ми переробимо його на приклад "2-2=0" (рис. 3.6).

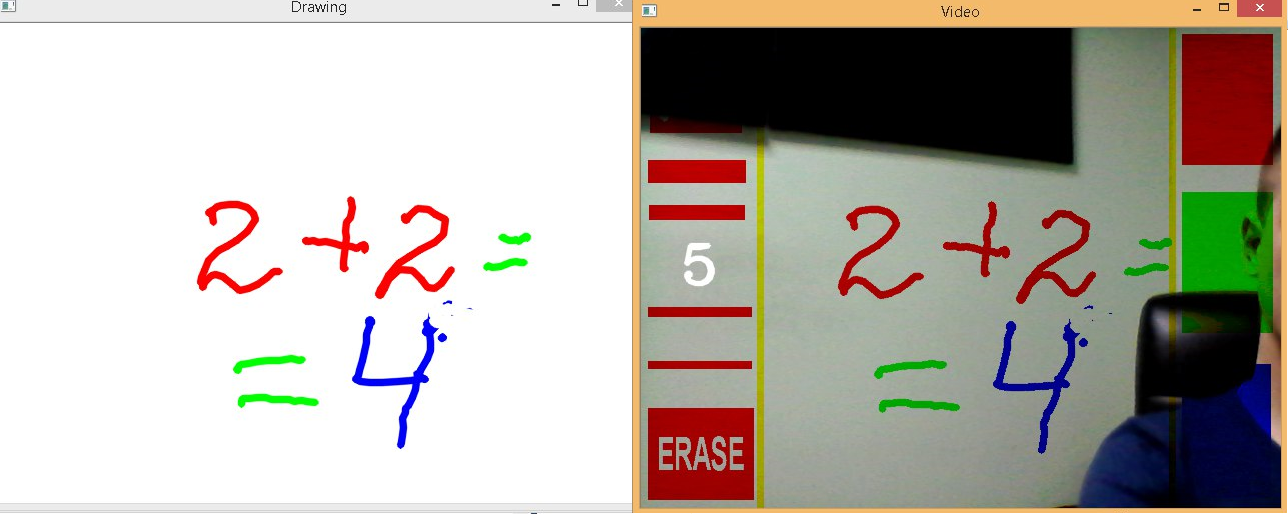


Рис. 3.5 Приклад додавання

Як бачимо, функція "гумки" працює. Також, передбачена функція повної очистки робочої поверхні, або "дошки". Для цього, користувачеві потрібно навести "перо" також у нижній лівий кут, та втримувати протягом 20 секунд, після чого вся робоча поверхня буде очищена (рис. 3.7). Відлік часу також буде показаний користувачу зверху по центру.

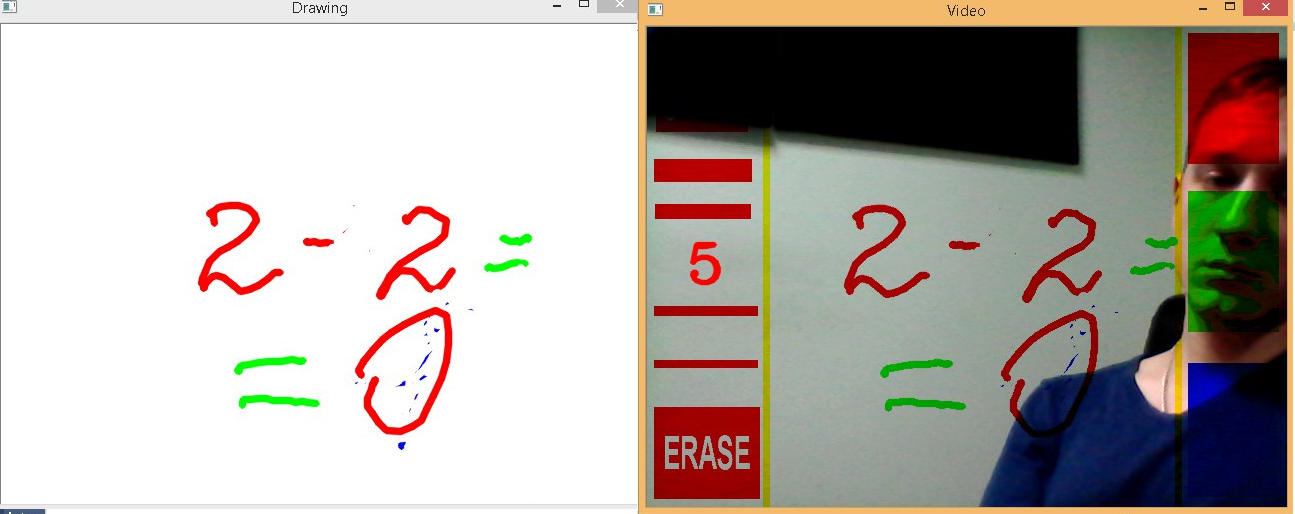


Рис. 3.6 Приклад віднімання

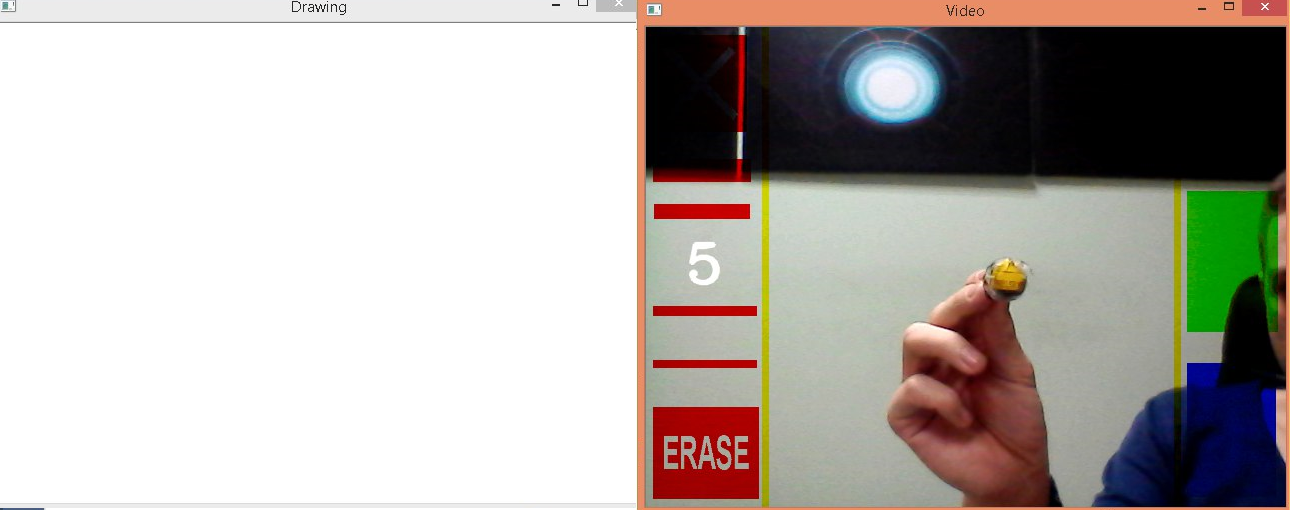


Рис. 3.7 Очищення робочої поверхні

## 3.2 Результати тестування програмної реалізації алгоритму

Згідно з поданого в розділі (2.1.) програмного коду було виконано тестування програми "відеоперо". У ході тестування був використаний звичайний ноутбук із вебкамерою, а також тестування проводилося на операційній системі «Windows 7 Максимальная».

При старті програми, перед користувачем з’являється три робочих вікна. В яких відображається зображення з вебкамери в режимі реального часу.

У вікні під назвою "Video" (рис. 3.8) відображається зображення з камери та "накладеним" інструментарієм користувача.

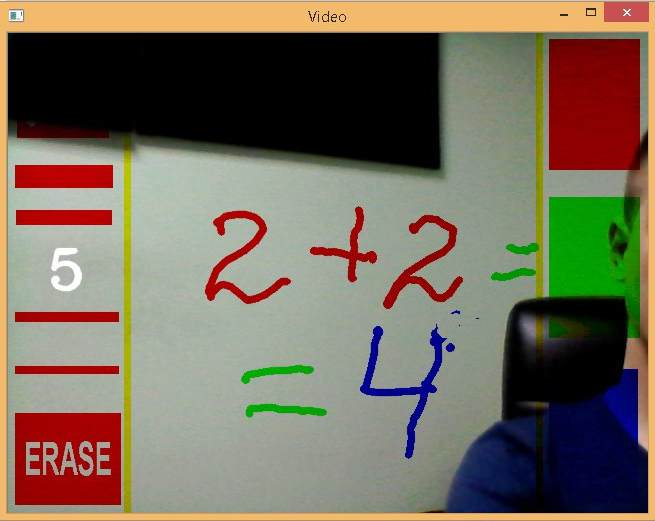


Рис. 3.8 Робоче вікно програми "Video"

У другому вікні за назвою "Drawing" - відображається результат роботи програми, а саме "дошка" із результатом дій користувача.

Для прикладу, був намальований приклад додавання "2+2=4". Для цього, були використанні три кольори та одну з найтовстіших ліній.

Також потрібно відмітите те, що функція з розпізнавання кольору "пера" має свої відмінності. Тобто, вона розпізнає тільки певний діапазон кольору, у нашому випадку - це жовтий. У даній функції, є певні недоліки. По перше - це погана освітленість приміщення. Тобто функція може банально не розпізнати колір об'єкту, тому що вона буде сприймати жовтий колір за темно-жовтий. Далі на вдале розпізнавання кольору, потрібно звертати уваги на якість вебкамери, яка підключена до комп'ютеру або ноутбуку. Вебкамера з низькою якістю сприйняття зображення може теж невірно розпізнати колір, або буде розпізнавати посередково, і в русі "пера" функція буде просто втрачати "перо".

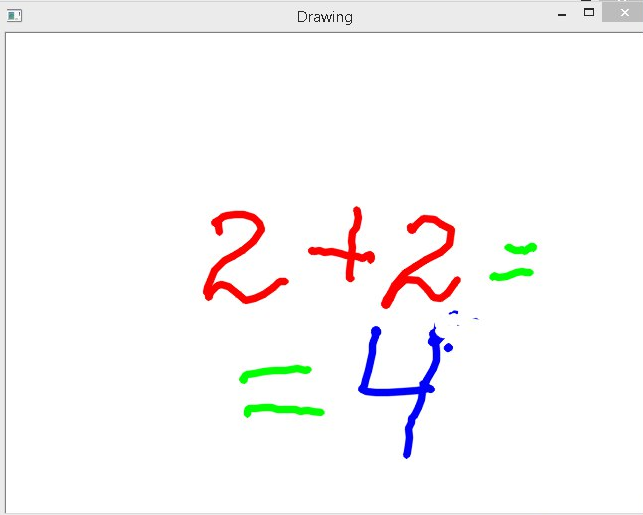


Рис. 3.9 Робоче вікно програми "Drawing"

Так у третьому вікні за назвою "Thershold" - відображається бінарне зображення пера, який у свою чергу обмальовується білим кольором (рис. 3.10). Дане вікно потрібно для того, щоб після запуску програмного забезпечення, перевірити вебкамеру та зображення на якість розпізнання об'єкту, та за змогою внести певні корективи до "пера", освітлення та самої камери. Також у якості "пера" може виступати абсолютно любий предмет, який має жовтий колір.

Для наявного прикладу, наведемо вебкамеру на предмети, з різними кольорами (рис. 3.11), і у вікні "Thershold" побачимо бінарне зображення предметів, які мають жовтий колір.

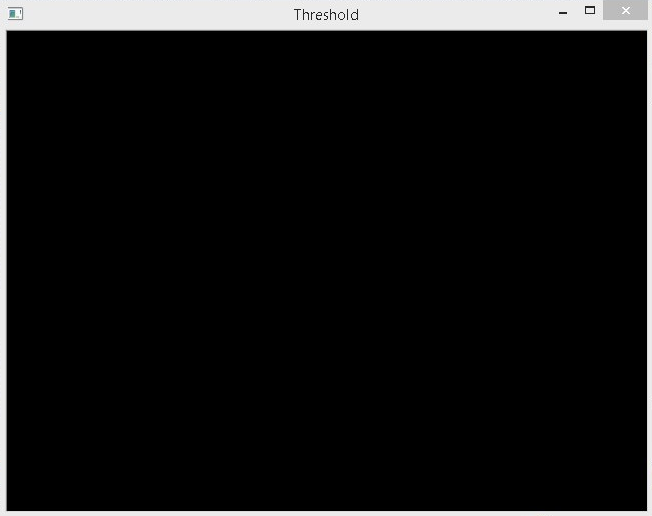


Рис. 3.10 Робоче вікно програми "Threshold"

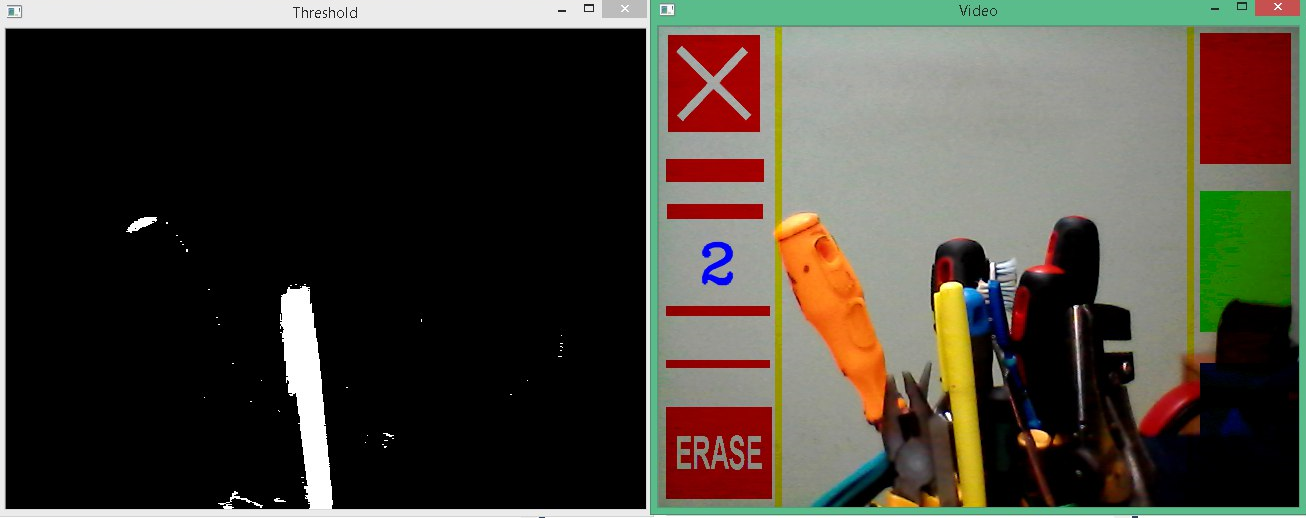


Рис. 3.11 Приклад визначення жовтого кольору

Як ми бачимо (рис. 3.11), у якості "пера" ми можемо використати маркер, який має зовні жовтий колір. Тобто на цьому прикладі, явно правильно підібраний предмет, а також освітлення та сама якість камери задовольняють наші потреби.

## 3.3 Висновки до розділу 3

1. Розроблений інтерфейс та інструкція користувача.

2. Проведено тестування програмного забезпечення відеопера мовою програмування С++ із використанням бібліотеки "OpenCV" на операційній системі Windows 7: Максимальна.

# ВИСНОВКИ

На основі виконаних досліджень властивостей системи комп'ютерного зору "OpenCV" та аналізу алгоритмів сегментації зображення й розпізнавання об'єктів була досягнута поставлена мета, та отримані такі результати:

1. Проведено аналіз наявних алгоритмів трекінгу об'єкта.

2. Створено оптимальний алгоритм із фрагментів "алгоритму пошуку об'єкта" , до яких добавлено елементи сервісного обслуговування доповідача на конференції, такі як вибір кольору пера, товщини ліній та "гумки".

3. Обґрунтовано вибір інструментів розробки програмного забезпечення відеопера для доповідача на науковій конференції . Виконана програмна реалізація оптимального алгоритму мовою програмування C++ із використанням бібліотеки "OpenCV". Програма дає надійні результати з визначення й детектування об’єкта певної форми, та застосування його в практичних цілях.

4. Розроблено інтерфейс та інструкцію користувача відеопера для доповідача на науковій конференції. Виконано тестування та корекцію створеного програмного забезпечення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Алфимцев А. Н. Метод обнаружения объекта в видеопотоке в реальном времени / Алфимцев А. Н., Лычков И. И. – Вестник ТГТУ, 2011 с. 44-55.
2. [Анисимов Б. В. Распознавание и цифровая обработка изображений: учебное пособие для студентов](http://www.computersbooks.net/index.php?id1=4&category=photoshop&author=anisimov-bv&book=1983) / Анисимов Б. В., Курганов В. Д., Злобин В. К. – М.: Высшая школа, 1998, с. 153-190.
3. Бакути П.А., Колмогоров Г.С. Ворновицька І.Е. Сегментація зображень: методи порогової обробки, 1999, с. 6-24.
4. Бакути П.А., Колмогоров Г.С. Сегментація зображень: методи виділення меж областей, 2000, c. 25-45.
5. Буэно, Суарес, Эспиноса. Обработка изображений с помощью OpenCV = Learning Image Processing with OpenCV. - М.: ДМК-Пресс, 2016. - 210с.
6. Венгерский алгоритм в задаче слежения за множеством движущихся объектов / Хабрахабр [Електроний ресурс] – Режим доступу: <https://habrahabr.ru/post/267651/>.
7. Гонсалес Р. В. Цифрова обробка зображень / Р. В. Гонсалес, Р. C. Вудс – М.: Техносфера, 2006, с. 1000- 1072.
8. Горелик А. Л., Скрипкин В. А. Методы распознавания. - 4е изд. - М.: Высшая школа, 2004, с. 123-190.
9. Денисов Д.А., Нізовкін В.А. Сегментація зображень на ЕОМ, 2001. №10, с. 5-31.
10. Желтов С.Ю. Обработка и анализ изображений в задачах машинного зрения, 2010. - 672с.
11. Журавель І.М. Короткий курс теорії обробки зображень: Межі зображень: Краї і їх виявлення.
12. Компьютерное зрение. Библиотека «OpenCV» [Електроний ресурс] / Анатолій Забровський – Режим доступу: http://itmultimedia.ru/kompyuternoe-zrenie-biblioteka-OpenCV/.
13. Кэлер А., Брэдски Г. Изучаем OpenCV 3 = Learning OpenCV 3. - М.: ДМК-Пресс, 2017. - 826с.
14. Л. Шапиро, Дж. Стокман. Компьютерное зрение - Computer Vision. - М.: Бином. Лаборатория Знаний, 2006, с. 620-750.
15. Л. Шапіро, Дж. Стокман; Комп'ютерний зір / Пров. з англ. - М .: БИНОМ. Лабораторія знань, 2006, c. 562-582.
16. Лукьяница А. А., Шишкин А.Г. Цифровая обработка видеоизображений. - М.: "Ай-Эс-Эс Пресс", 2009. - 518с.
17. Махфуд У. А. «Комбіновані алгоритми сегментації кольорових зображень»: дис. канд. тех. наук: 05.13.01 / У. А. Махфуд; Інститут технічної кібернетики Національної академії наук Білорусі. – Мінськ, 2002.
18. Пару слов о распознавании образов / Хабрахабр [Електроний ресурс] – Режим доступу: https://habrahabr.ru/post/208090/.
19. Распознавание образов для программистов [Електроний ресурс] / Александр Кручинин – Режим доступу: <http://blog.vidikon.com/?cat=3>.
20. Фомин Я. А. Распознавание образов: теория и применения. - 2-е изд. - М.: ФАЗИС, 2012. - 429с.
21. Фомин Я. А., Тарловский Г. Р. Статистическая теория распознавания образов. - М.: Радио и связь, 1999. - 624с.
22. Форсайт Дэвид А., Понс Джин. Компьютерное зрение. Современный подход - Computer Vision: A Modern Approach. - М.: Вильямс, 2004. - 928с.
23. Хорн Б.К.П. Зір роботів: Пер. з англ. - М .; Світ, 2003. - 487 с., Іл.
24.  Хуанг Т. М. Обробка зображень і цифрова фільтрація / Т. М. Хуанг – М.: Мир, 2004, с. 253-300.
25. Шапіро Л. Машинний зір / Л. Шапіро, Дж. Стокман; Пров. з англ. – М.: Біном. Лабораторія знань, 2006, с. 592-620.
26. Яне Б. П. Цифрова обробка зображень / Б. П. Яне – М.: Техносфера, 2007, с. 233-291.
27. Ярославский Л. П. Введение в цифровую обработку изображений / Леонид Пинхусович Ярославский. – М.: Советское радио, 2006, с. 290- 312.
28. Ярославський Л. Введення в цифрову обробку зображень, Глава - Обробка зображень, 2001, с. 156-252.
29. Gary Bradski, Adrian Kaehler Learning OpenCV: Computer Vision with the OpenCV Library, 2008. - 580.
30. Robert Laganiere Computer Vision Application Programming Cookbook, 2011. - 287.