

Вінницький національний технічний університет  
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматики  
(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук  
(повна назва кафедри (предметної, циклової комісії))

**Магістерська дипломна робота на тему:**

**«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РЕЙТИНГУВАННЯ ФОТОАПАРАТІВ ЗА ЯКІСТЮ. ЧАСТИНА 1. ОЦІНЮВАННЯ РІВНЯ ШУМУ ФОТОАПАРАТІВ»**

Виконав: студент 4 курсу, групи 1КН-22м  
спеціальності 122 – Комп'ютерні науки  
(шифр і назва напрямку підготовки, спеціальності)

Михайленко Є.О.  
ИНА(прізвище та ініціали)

Керівник: к.т.н., доцент кафедри КН

Сілагін О.В.  
(прізвище та ініціали)

«07» 12 2023 р.

Опонент: к.т.н., доцент кафедри АІТ

Богач І.В.  
(прізвище та ініціали)

«07» 12 2023 р.

Допущено до захисту  
Зав. кафедри

«08» 12 2023 р.

Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та автоматизації  
Кафедра Комп'ютерних наук  
Рівень вищої освіти – перший (бакалаврський)  
Галузь знань – 12 «Інформаційні технології»  
Спеціальність – 122 «Комп'ютерні науки»  
Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ

Завідувач кафедри КН

проф., д.т.н. Яровий А. А.

«29» 08 2023р.

### ЗАВДАННЯ

#### НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

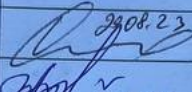
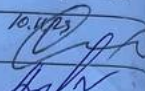
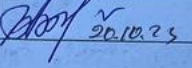
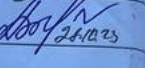
Михайленку Євгенію Олександровичу

1. Тема роботи: Інформаційна технологія рейтингування фотоапаратів за якістю. Частина 1. Оцінювання рівня шуму фотоапаратів  
керівник роботи: Сілагін О.В. к.т.н., доц. каф. КН  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)  
затвержені наказом вищого навчального закладу «18» 09 2023 року № 247
2. Термін подання студентом роботи: 13.11.2023 р.
3. Вихідні дані до роботи: інтегральний показник якості оцінює показники кольороспотоворення, шуму та дисторсії; частинний показник величини шуму оцінюється в режимі «АВТО» при зміні ISO та освітлення; для оцінювання якості використати 12-ти рівневу шкалу лінгвістичних термів; дзвіноподібна функція належності; формати графічних файлів – JPEG, PNG, RAW; підтримка загального інтерфейсу баз даних ODBC;
4. Зміст текстової частини (перелік питань, які потрібно розкрити): Аналіз методології оцінювання шумів цифрових фотокамер, огляд існуючих способів вирішення проблеми вимірювання рівня шумів, розробка математичної моделі тестування на шумові завади, оцінити рівні шумів методом нечіткої логіки,

розробити алгоритм тестування цифрових фотоапаратів на шумові завади, розробити спрощений програмний модуль для роботи з нечіткими знаннями.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): Граф виведення результатів оцінювання. Алгоритм тестування на шумові завади, загальна схема алгоритму роботи програми, результати роботи програми.

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3	Сілагін О.В., к.т.н., доц. каф.КН	 29.08.23	 10.10.23
4	Адлер О.О., доц. каф. ЕПВМ	 20.10.23	 21.10.23

7. Дата видачі завдання 29.08.2023 р.

#### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз існуючих інформаційних технологій оцінювання рівня шуму	01.09.23-07.09.23	
2	Математичне моделювання оцінювання рівня шуму та розробка технології	08.09.23-15.10.23	
3	Проектування та реалізація програмного модуля оцінювання шуму	16.10.23-20.10.23	
4	Оформлення пояснювальної записки і графічного матеріалу	21.10.23-28.10.23	
5	Попередній захист	30.10.23-05.11.23	
6	Остаточний захист	06.11.23-10.11.23	

Студент

(підпис)

Михайленко С.О.  
(прізвище та ініціали)

Керівник проекту (роботи)

(підпис)

Сілагін О.В.  
(прізвище та ініціали)

## АНОТАЦІЯ

УДК 004.9:656.1

Михайленко Є. О. Інформаційна технологія рейтингування фотоапаратів за якістю. Частина 1. Оцінювання рівня шуму фотоапаратів. Магістерська кваліфікаційна робота зі спеціальності 122 Комп'ютерні науки, освітня програма – Системи штучного інтелекту. Вінниця: ВНТУ, 2023. 99 с.

На укр. мові. Бібліогр.: 24 назв; рис.: 12; табл. 6.

Магістерська кваліфікаційна робота входить до складу комплексної магістерської роботи і присвячена оцінюванню рівня шуму фотоапаратів у рамках розробки інформаційної технології рейтингування фотоапаратів за якістю. Аналізуються всі можливі методи виявлення та вимірювання шумових спотворень. Величину спотворення пропонується оцінювати у вигляді нечіткої множини, що відповідає лінгвістичному терму за певною функцією належності. Для логічного виведення оцінки застосовується нечітка база знань та механізм ієрархічного нечіткого логічного виводу.

У розділі економічної частини здійснено оцінювання комерційного потенціалу розробки інформаційної технології рейтингування фотоапаратів за якістю, проведено оцінювання комерційного потенціалу розробки, спрогнозовано витрати на виконання наукової роботи та впровадження результатів, розраховано період окупності.

Ілюстративна частина складається з 6 плакатів із результатами.

Ключові слова: рейтингування, оцінка, шуми, нечітка логіка.

## ABSTRACT

Mykhaylenko E. O. Information technology for rating cameras by quality. Part 1. Evaluation of the noise level of cameras. Master's thesis on the specialty 122 Computer science, educational program - Artificial intelligence systems. Vinnytsia: VNTU, 2023. 99 p. In Ukrainian speech Bibliography: 24 titles; Fig.: 12; table 6.

The master's qualification work is part of the complex master's work and is devoted to the evaluation of the noise level of cameras as part of the development of information technology for rating cameras by quality. All possible methods of detecting and measuring noise distortions are analyzed. The amount of distortion is proposed to be estimated in the form of a fuzzy set corresponding to a linguistic term according to a certain membership function. For the logical derivation of the estimate, a fuzzy knowledge base and a mechanism of hierarchical fuzzy logical inference are used.

In the section of the economic part, an assessment of the commercial potential of the development of information technology for rating cameras by quality was carried out, an assessment of the commercial potential of the development was carried out, the costs of carrying out scientific work and the implementation of the results were forecast, and the payback period was calculated.

The illustrative part consists of 6 posters with results.

Keywords: ranking, assessment, noise, fuzzy logic.

## ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ІСНУЮЧИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ОЦІНЮВАННЯ РІВНЯ ШУМУ.....	7
1.1 Аналіз методології оцінювання шумів цифрових фотоапаратів.....	8
1.2 Існуючі способи вирішення проблеми вимірювання шумів, огляд аналогів та обґрунтування вибору.....	12
1.3 Технічні вимоги до системи оцінювання шуму растрових зображень.....	14
1.4 Висновок до розділу 1.....	15
2 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ОЦІНЮВАННЯ ФОТОКАМЕР ТА РІВНЯ ШУМУ.....	16
2.1 Обґрунтування та вибір методології моделювання.....	16
2.2 Принципи лінгвістичного моделювання.....	16
2.3 Постановка задачі моделювання оцінки якості растрових зображень.....	20
2.4 Моделювання оцінювання якості растрових зображень як ієрархічний об'єкт.....	21
2.5 Представлення баз знань нечіткими логічними рівняннями.....	29
2.6 Розробка алгоритму виведення узагальненої оцінки.....	31
2.7 Формалізація моделі узагальненого логічного виводу оцінки якості фотокамери...32	32
2.8 Розробка діаграми класів програмного забезпечення системи.....	33
2.9 Математична модель тестування на шумові завади.....	36
2.10 Визначення рівня шумів на основі шумів медіанної фільтрації.....	37
2.11 Оцінювання рівня шумів методом нечіткої логіки.....	42
2.12 Модель оцінки інтегрального показника.....	44
2.13 Висновок до розділу 2.....	46
3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РЕЙТИНГУВАННЯ ФОТОАПАРАТІВ ЗА ЯКІСТЮ. ОЦІНЮВАННЯ РІВНЯ ШУМУ.....	47
3.1 Розробка алгоритму тестування цифрових фотоапаратів на шумові завади.....	47
3.2 Реалізація програмного модуля вимірювання та оцінки рівня шумів.....	49

3.3 Розробка алгоритму роботи модуля оцінки якості растрових графічних зображень.....	51
3.4 Висновок до розділу 3.....	54
4 ЕКОНОМІЧНА ЧАСТИНА.....	55
4.1 Проведення комерційного та технологічного аудиту інформаційної технології рейтингування фотоапаратів за якістю та оцінювання рівня дисторсії та кольороспотворення.....	55
4.2 Розрахунок витрат на здійснення науково-дослідної роботи.....	56
4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором.....	62
4.4 Висновок до розділу 4.....	66
ВИСНОВКИ.....	67
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	68
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	70
Додаток Б (обов'язковий) Лістинг програми.....	71
Додаток В (обов'язковий) Ілюстративна частина.....	92
Додаток Г (довідниковий) Інструкція користувача.....	99

## ВСТУП

### Актуальність теми

Присутність в зображеннях цифрових фотокамер шумів є серйозною проблемою як для величезної армії фотолюбителів так і, тим більше, професіоналів. Як правило, чим менше «професіоналізм» фотокамери, її якість, і відповідно її ціна, тим більше рівень завад, пов'язаних із шумами та дисторсією. Звичайно любитель-початківець і фотограф–професіонал по різному формують уявлення про якість своїх фотокамер. Іншими словами оцінку якості фотокамер треба робити відносно до певного цінового або користувацького класу, і тільки автоматизована оцінка якості дасть можливість усунути суб'єктивну складову.

Оцінку якості цих пристроїв необхідно робити при виробництві самих фотокамер, їх тестуванні, налаштуванні, а також в торгівлі і в процесі експлуатації.

Якість фотоапаратів можна визначати за величиною спотворень які вони вносять при відтворенні, наприклад, тестового зображення. Традиційно аналізуються три види спотворень - спотворення кольору, цифровий шум, та геометричні спотворення. Пропонована робота присвячена розробці частини інформаційної технології, що аналізує саме шумові спотворення. Оцінювання рівня спотворень проводять порівнюючи цифрові зображення спеціальних стандартизованих тестів з їх електронними аналогами. Власне оцінювання відбувається з використанням ймовірностних та Байєсовських моделей, а в останній час, ще й з використанням інтелектуальних технологій, пов'язаних з використанням нейронних мереж, генетичних алгоритмів та нечіткої логіки.

Практика використання систем оцінки якості, заснованих на ймовірностних моделях показала деяку їх недостатність і потребу врахування психо-фізіологічних особливостей людини. У роботі показано доцільність використання для подібних підходів методології експертних систем заснованих на базах знань, але при цьому ми маємо поєднання двох типів формалізації невизначеності: «ймовірно» та «можливо» що робить кінцеву модель достатньо



громіздкою, а при взаємних перетвореннях призводить до втрати адекватності. Пропонується відмовитись від формалізації невизначеності через функцію розподілення теорії ймовірності і замінити її функцією належності із теорії нечітких множин. Дана робота є другою частиною комплексної магістерської дипломної роботи по створенню інформаційної технології автоматизованої об'єктивної оцінки якості тому дослідження по цій темі вважаю актуальними.

### **Зв'язок роботи з науковими програмами, планами, темами**

Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету та плану наукової та навчально-методичної роботи кафедри.

### **Мета дослідження**

Метою даного дослідження є підвищення достовірності оцінки рівня шуму на растрових графічних зображеннях для подальшого ринкового рейтингування фотокамер за якістю.

### **Задачі дослідження**

Для досягнення поставленої мети необхідно розв'язати такі задачі подальшого дослідження:

- аналіз існуючої методології вимірювання шумів;
- математичне моделювання та формалізація ієрархічного процесу виведення узагальненого (інтегрованого) показника рівня шумів;
- математичне моделювання вимірювання рівня шумів;
- проектування та реалізація алгоритма оцінювання рівня шуму.
- проектування та реалізація програмного модуля технології оцінювання якості фотокамер;
- Провести тестування програмного продукту та виконати аналіз отриманих результатів;
- Розробити інструкцію користувача.
- Розробити економічну частину.

**Об'єктом дослідження** є процес оцінки зашумленості растрових зображень з використанням інформаційних технологій для подальшого рейтингування фотоапаратів за якістю.

**Предметом дослідження** математичні моделі та програмні засоби вимірювання рівня шумів на растрових графічних зображеннях.

**Наукова новизна одержаних результатів.**

Розроблено інформаційну технологію оцінювання якості фотокамер, що відрізняється від існуючих поєднанням класичних методів вимірювання величини спотворень растрових зображень з методами оцінювання на основі апарату нечіткої множини, що дозволяє підвищити достовірність параметрів рейтингування.

**Практичне значення отриманих результатів** виявляється в тому, що розроблено програмний модуль, що дозволяє автоматизувати процес оцінки рівня шуму у растрових графічних зображеннях та рейтингування фотоапаратів за якістю. Завдяки цьому, достовірність отриманих результатів матиме достатньо високий рівень.

**Достовірність теоретичних положень** магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

**Особистий внесок здобувача.** Усі результати, що наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, які написано у співавторстві, здобувачу належать: алгоритм оцінки рівня шумів.

**Апробація результатів роботи.** Результат роботи були апробовані на ЛІІ науково-технічній конференції «Молодь в науці» ВНТУ в 2023 році [1].

**Публікації.** За результатами роботи опубліковані тези доповідей на науково-технічній конференції «ЛІІ Науково-технічна конференція Молодь в науці» (м. Вінниця, Україна) [1] у 2023 році.

## 1 АНАЛІЗ ІСНУЮЧИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ОЦІНЮВАННЯ РІВНЯ ШУМУ

Досить важко уявити собі ту сферу інформаційних технологій, або навіть сферу людської діяльності де б не використовувались сфотографовані або відскановані растрові графічні зображення. Величезна кількість виробників наращують випуск та розширюють асортимент різноманітних пристроїв для одержання цих зображень. Звичайно, важко порівняти між собою фотокамеру для аерокосмічної зйомки з дешевою «мільницею, або барабанний професійний видавничий сканер з побутовим планшетним. Їх характеризує не тільки величезна різниця у вартості, а, в першу чергу, якість одержаних за їхньою допомогою растрових зображень. Звичайно, будь який споживач подібних пристроїв прагне придбати потрібну йому «якість» (набір потрібних йому якісних показників) за мінімальною ціною. Це спонукає недобросовісних виробників завищувати якісні показники своєї продукції, щоб підвищити її конкурентоспроможність. Крім того дуже актуальною є задача періодичного тестування пристроїв для одержання растрових зображень в процесі їх експлуатації, так як показники якості з часом погіршуються, в залежності від умов та інтенсивності експлуатації. Все вище перераховане породжує серйозну технічну проблему в галузі виробництва, торгівлі та експлуатації пристроїв для одержання растрових зображень, яку можна сформулювати як проблему об'єктивної оцінки якості, як по окремим якісним показникам, так і по деякому інтегрованому показнику в цілому. Що ж це за показники?

По перше саме поняття якості пристроїв для одержання растрових графічних зображень (і відповідно самих растрових зображень) ми можемо сформулювати як деяку ступінь спотворення тестового (ідеального) зображення. Ці спотворення традиційно ділять на три досить незалежні одна від одної групи:

- кольороспотворення;
- шумові спотворення (шуми);
- дисторсія (геометричні спотворення);

Останні 2 і є предметом дослідження даної другої частини комплексної магістерської дипломної роботи.

### **1.1 Аналіз методології оцінювання шумів цифрових фотоапаратів**

В характеристиках цифрових фотоапаратів ніколи не вказується один дуже важливий параметр, що виникає під час фотографування – це наявність шумів на знімку [2]. На деяких цифрових знімках іноді замість однорідного тону видно кольорові смужки, такі перекручування називаються «шум». Шуми проявляються не як збільшення пікселів (крапок, з яких складається зображення), а як поява на зображенні об'єкта неоднорідних по яскравості й кольорам пікселів.

Шуми, що виникають на знімку, залежать від характеристик матриці фотокамери та режимів фотографування, тобто інтенсивності освітлення матриці. Теоретично, чим більша матриця, тим менші шуми, чим менше освітлення, тим більше шумів відносно корисного сигналу.

На виникнення шумів також впливає ISO [2], тобто чутливість матриці. Найкращу якість знімку можна отримати при мінімальній чутливості, при виборі високої чутливості збільшується рівень шумів і контраст зображення, звужується динамічний діапазон.

Матриця сучасної камери – це цифровий аналог плівки. Невелика мікросхема прямокутної форми, на яку через об'єктив проектується зображення. Датчики сенсора – пікселі, перетворювачі світла в електричний сигнал, грають тут ключову роль. Чим більше фізичний розмір пікселя, тим більша площа, що поглинає світло і тим нижче рівень шумів на знімку. До того ж потрібно враховувати, що під час роботи датчики нагріваються, що також загрожує появою шумів. Саме через це шуми виявляються при зйомці на довгих витримках.

Шуми проявляються на знімку у вигляді хаотично розподілених кольорових плям. Особливо сильно спостерігаються шуми в тінях. Також шум

може виявлятися у вигляді зернистості, майже як на високочутливої плівці – такого роду спотворення майже не впливають на кольоровість і контраст фотографії і дратують найменше (якщо, звичайно, ця зернистість не притаманна всім без винятку знімкам).

При виборі камери число мегапікселів часто буває переконливим аргументом на користь певної моделі. Однак не варто гнатися за мегапікселями, оскільки розміри пікселів і сенсора, відповідно, значно важливіші. Адже чим більше пікселів розміщено на матриці, тим менше їх розмір і тим щільніше вони розташовані. Щільно «упаковані» датчики швидко нагріваються, і шуми можуть з'явитися вже не на мінімальній чутливості і при коротких витримках.

Найбільш низький рівень шумів у повноформатних дзеркальних камер з розміром сенсора 36x24 мм (як у кадрі 35мм плівки). Це такі камери як Canon EOS Mark II 1Ds (16 Мп), Canon EOS 5D (12 Мп). Вони прекрасно справляються з ISO 800, і навіть при чутливості ISO 1600 зроблені ними знімки виглядають дуже непогано.

Матриця зеркалок, таких, як, наприклад, Canon EOS 20D, менше плівкового кадру, вона має розміри 22,5 x15 мм, що також дозволяє знімати на досить високій чутливості. Цей формат сенсора отримав назву APS-C. Звичайно, вони поступаються камерам з повноформатної матрицею, але в порівнянні з будь-якими компактними камерами результат буде набагато краще.

На сьогоднішній день єдина в світі камера з незнімним об'єктивом, яка має велику матрицю формату APS-C – це Sony R1. У неї рівень шумів при високій світлочутливості можна порівняти з рівнем дзеркальних камер.

У переважної більшості компактних камер шуми дають про себе знати вже при чутливості від ISO 200 і вище. Так що, незважаючи на можливість встановити ISO 800, 1600 чи навіть 3200, можете забути про ці значення, так як картинка швидше за все буде просто непридатною для використання. Шуми, навіть при короткій витримці, стають настільки сильними, що про деталізацію і вірної передачі кольору можна забути. Заявлена виробником висока чутливість на практиці найчастіше виявляється просто неробочою.

Однак не всі сенсори «не дружать» з високою чутливістю. У камер Fujifilm сенсори виготовлені за власною унікальною технологією. Завдяки конструктивним особливостям матриці, рівень шумів дійсно помітно знижений у порівнянні з іншими типами сенсорів при тих же фізичних розмірах. Чим це пояснюється?

Традиційна фотоплівка містить кристали галогеніда срібла різних форм – від високочутливих зерен з великою площею поверхні, чутливих навіть до дуже слабкого світла, і зерен з маленькою площею поверхні, що реагують на яскраве світло. Технологія Super CCD SR пропонує аналогічне «поділ праці», об'єднуючи інформацію з низькочутливих і високочутливих пікселів. Розроблена Fujifilm матриця поєднує в собі великі високочутливі S-пікселі і більше маленькі, відповідно менш чутливі, R-пікселі, що розширює динамічний діапазон. Завдяки цьому камери Fujі непогано справляються з високими значеннями ISO – 400, 800 одиниць.

А ось новітня камера FinePix S6500fd дозволяє змінювати чутливість від ISO 100 до ISO 3200. Звичайно, така висока чутливість як 3200 навряд чи забезпечить вам прийнятну картинку. Але сумніватися в тому, що камера чудово відпрацює на ISO 400, не доводиться.

Але це тільки у камер Fujі! У решти - висока світлочутливість – дуже слабке місце. Наприклад, у популярного ультразума Sony H5 максимально заявлена чутливість ISO 1000. Але реально на практиці камера починає шуміти вже при ISO 400, і при більш високих значеннях знімок точно втратить будь-яку привабливість. Точно так само шуми помітні при установці ISO 400 і у Canon PowerShot S2 IS. У багатьох випадках спалах допомагає хоч якось боротися з цією проблемою.

Ультракомпакти мають найгірші характеристики в любительському класі. По-перше, у них матриця має найменші розміри. Крихітний сенсор шумить, природно, дуже сильно. По-друге, оптика у них також вкрай мініатюрна, що теж не сприяє підвищенню якості зображення. Хоча, звичайно, на шуми оптика не впливає, за це відповідальний тільки сенсор.

Деяка перевага – у камер з оптичною стабілізацією. Якщо в камері (або в знімному об'єктиві зеркала) є оптична стабілізація, то це дозволяє при нестачі освітлення збільшити витримку і не підвищувати значення ISO. Але оптичний стабілізатор рятує тільки на дві, максимум три сходинки. І якщо у камери маленький сенсор, то тут навіть стабілізація не дуже допоможе. Яскравий приклад – серія камер Panasonic Lumix FZ. Наявна сьогодні на озброєнні в багатьох фотолюбителів FZ30 в цілому дуже хороша камера, що має масу переваг: відмінний дизайн, ергономіка, функціональність, висока деталізація, перенесення кольорів і т.д. Але головний мінус цієї прекрасної стильної камери – її шуми. Вони дуже помітні на зображенні навіть при низькій чутливості, не кажучи вже про ISO 400, коли шуми роблять зображення просто неприйнятним для використання.

Всілякі системи шумозаглушення, вбудовані в фотокамеру, не рятують від шумів. Зернистість вони, звичайно, згладжують, але при цьому погіршується деталізація, спотворюється передача кольору. Картинка виглядає неприродною, «підфарбованою». Реально якісного зображення на високій чутливості при такому підході ви не отримаєте. Краще недотримати кадр на 1-1,5 шаблі, а потім його «витягнути» при обробці в графічному редакторі, ніж отримати яскраву і, в кінцевому підсумку, марну картинку.

Певні переваги дає зйомка в форматі RAW, оскільки цей формат містить більше інформації. Знімок не проходить постцифрову обробку, власне, тому цей формат і називається сирим. Знімки, записані в RAW, набагато простіше «витягнути» в спеціальному конверторі.

Формат JPEG передбачає стиснення з втратами і вносить спотворення до зображення. Частина деталей при цьому втрачається. При збереженні в найвищій якості спотворення майже не помітні на око, але при подальшому редагуванні і збереженні файлу він ще стискається, дані знову губляться. Якщо передбачається обробка JPEG-файлу (наприклад, потрібно прибрати шуми), то варто переконвертувати файл JPEG в графічний формат, що не вносить спотворень у вихідну картинку, наприклад TIFF.

## 1.2 Існуючі способи вирішення проблеми вимірювання шумів, огляд аналогів та обґрунтування вибору

Як уже говорилося вище, на величину шумів впливає багато різноманітних чинників серед яких, характер зовнішнього освітлення, якість та технологія виготовлення фотооптики, розмір та конструкція виготовлення світлочутливих матриць, режими фотозйомки і тд. Можна виділити дві методології оцінки величини шумів.

Перша - це математичне моделювання [2]. Можна створити математичні моделі проходження та перетворення світла різних оптичних систем, матриць, підсилювачів та перетворювачів і на цих моделях виявляти шуми в різних режимах. Такий підхід до оцінки якості зображень полягає у використанні апріорної інформації про викривлення, що виявляються при формуванні, кодуванні або передачі візуального сигналу. У разі обмеженості доступних ресурсів візуальна інформація піддається різного типу спотворенням: блочність, шумоспотворення.

Якщо система обробки така, що вихідне зображення містить лише кінцеве число різних типів спотворень, то незалежна оцінка кожного виду спотворень дозволяє судити про якість всієї системи. Перевага такого підходу перед описаними вище полягає в тому, що навіть сильно корельовані між собою викривлення і спотворення, залежні від вхідного зображення, вимірюються незалежно. Це дозволяє уникнути ситуацій, коли різні за якістю зображення не відрізняються з точки зору міри близькості їх до оригіналу. Більш того, незалежні метрики кількісної оцінки спотворень можуть бути використані для оптимізації параметрів пристроїв з метою підвищення якості.

Адекватне вимірювання і аналіз спотворень дозволяє розробляти алгоритми їх усунення і поліпшення якості відновлених зображень. Переваги цієї методології в швидкості отримання результатів(після того як створена та відтестована базова модель). Серйозним недоліком цієї групи є надзвичайно вузька спеціалізація створеної моделі для певного типу оптичної системи, або матриці,, що



застосовується в пристроях для одержання растрових зображень і, оскільки, проведення досліджень вимагає роботи в спеціалізованих середовищах для математичного моделювання, то і, відповідно, високої кваліфікації дослідників. Таким чином методи цієї групи, як правило, застосовуються в наукових установах для проведення наукових досліджень.

Друга методологія – емпірична, полягає в реальному (фізичному) скануванні тестового зображення і порівняння його з ідеальним цифровим «еталоном» є найпоширенішою і найбільш застосованою. Прикладом емпіричного методу вимірювання кольороспостворення є система описана в [3]. До переваг цієї методології слід віднести її універсальність. Вона полягає в тому, що система створена на основі емпіричного підходу може використовуватись для широкої гамми пристроїв і вимірювати майже всі види кольороспостворенняв єдиному інтегрованому показнику. Серед недоліків цієї методології потрібно відмітити:

- дороге технологічне обладнання;
- участь у процесі вимірювання кваліфікованого персоналу;
- трудомісткий та тривалий процес вимірювання і обробки результатів.

При вимірюванні шумів використовують також гістограми яскравості зображення [4]. Гістограма – це один зі стандартних графіків, які використовуються для візуального представлення масиву математичних даних. Гістограма яскравості зображення – це і є така стовпчаста діаграма, що відображає кількість пікселів зображення (по вертикалі), що мають заданий рівень яскравості (по горизонталі). Вона побудована на основі “таблиці”, що являє собою матрицю зі світлочутливих елементів у цифровій фото камері [5]. Тобто масивом даних є зображення, в якому кожен піксель матриці передає значення яскравості світла, що надійшло на нього в кожному з кольорових каналів.

Гістограма яскравості являє собою графік, що показує, які відтінки присутні в зображенні. Діапазон відтінків зображення представлений у вигляді послідовності вертикальних ліній, розташованих з ліва на право від самого темного до самого світлого. Висота кожної лінії показує, скільки в зображенні

пікселів відповідного відтінку. Гістограму можна одержати й для кожного кольорового каналу окремо: червоного (Red), зеленого (Green), синього (Blue), блакитного (Cyan), пурпурного (Magenta), жовтого (Yellow), чорного (black) та для Lab. У цьому випадку вона показує кількість пікселів, що мають певне значення величини яскравості для кожного кольору. Величина яскравості змінюється в діапазоні від 0 до 255 з точністю до одиниці, що відповідає колірній глибині в 8 біт на кожен кольоровий канал [6].

Гістограма є допоміжною функцією для оцінки якості фотографії, тому одержувану з її допомогою інформацію потрібно співвідносити із зображенням на дисплеї цифрової камери. На гістограмі розподілені ділянки зображення які викликають виникнення смуг. Ліворуч на графіку відображаються темні ділянки знімка, тобто тіні. Праворуч – світлі ділянки. Якщо підйоми розташовані приблизно симетрично, ближче до центра гістограми, то зображення добре збалансовано. Якщо підйоми зміщені вліво або вправо щодо центра, то це є показником того, що зображення, відповідно, занадто темне або світле [6].

### **1.3 Технічні вимоги до технології оцінювання шуму растрових зображень**

У відповідності до матеріалу, що викладений у розділі 1.2 ми можемо сформулювати вимоги до системи вимірювання та оцінки якості шумових спотворень:

Функціональні вимоги:

- процес вимірювання повинен виконуватись автоматизовано (інтерактивний режим) після встановлення пристрою для одержання растрових графічних зображень «фотокамера»), тестового зображення, підключення апаратної частини інтерфейсу та налаштування користувацького інтерфейсу;
- система повинна мати два режими роботи «сканер» та «фотокамера»;
- в режимі «фотокамера» система повинна вимірювати хроматичну аберацію об'єктива і оцінювати її рівень в термах нечіткої логіки;

- в режимі «фотокамера» система повинна вимірювати аберацію «кома» об'єктива і оцінювати її рівень в термах нечіткої логіки;
- в режимі «фотокамера» система повинна вимірювати астигматизм оптичної системи і оцінювати його рівень в термах нечіткої логіки;
- в режимі «фотокамера» система повинна вимірювати дисторсію оптичної системи і оцінювати її рівень в термах нечіткої логіки;
- оцінка геометричних спотворень в термах нечіткої логіки повинна мати 5 рівнів (дуже погано, погано, задовільно, добре, дуже добре);
- система користувацького інтерфейсу повинна бути реалізована у вигляді ієрархічних «меню» із «спливаючими вікнами» та «полосами прокручування»;
- система повинна формувати інтегрований показник оцінки якості пристрою на геометричні спотворення оцінювати його рівень в термах нечіткої логіки;
- система повинна накопичувати результати вимірювання пристроїв для одержання растрових графічних зображень («сканер», «фотокамера») у власній базі даних і на основі цієї інформації коригувати шкалу оцінки якості в термах нечіткої логіки.

#### **1.4 Висновок до розділу 1**

В першому розділі МКР проведено огляд та аналіз існуючих технологій методів та алгоритмів вимірювання та оцінювання шумів та дисторсії, а саме:

- аналіз методології оцінювання шумів цифрових фотокамер;
- існуючі способи вирішення проблеми вимірювання шумів. Огляд аналогів та обґрунтування базового;
- аналіз методології вимірювання дисторсії; існуючі способи вимірювання дисторсії. Огляд аналогів та обґрунтування базового.

## 2 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ОЦІНЮВАННЯ РІВНЯ ШУМУ

### 2.1 Обґрунтування та вибір методології моделювання

Задача ідентифікації рівня якості (як і будь яка задача ідентифікації), як правило, вирішується у два етапи. На першому, що називається структурною ідентифікацією, складається початкова модель об'єкту, яка апроксимує взаємозв'язок вхід-вихід та має налагоджувані параметри. На другому етапі, що називається параметричною ідентифікацією, добираються такі значення параметрів, які мінімізують відстань між модельним та експериментальним виходами об'єкту. Етап структурної ідентифікації уявляє собою скоріше мистецтво, ніж науку, оскільки вибір настроюваної моделі залежить від самого пошукача: його освіти, розуміння суті об'єкту, пристрасті до того або іншого математичного апарату та інших суб'єктивних причин. У якості настроюваних моделей у сучасній теорії ідентифікації використовуються кількісні співвідношення у вигляді різних рівнянь: алгебраїчних, диференціальних, інтегральних та інших.

Етап параметричної ідентифікації достатньо формалізований, так як зводиться до застосування різних методів оптимізації. Складність виявляється лише у пошуку глобального мінімуму нелінійних функцій відхилення теорії від експерименту, та ускладненню обчислень при збільшенні кількості налагоджуваних параметрів.

### 2.2 Принципи лінгвістичного моделювання

Сформулюємо ряд принципів [14] котрі будуть використані нами при розробці методу ідентифікації нелінійних об'єктів на основі нечітких баз знань. Ці принципи являються узагальненням і подальшим розвитком аналогічних принципів, сформульованих для прийняття діагностичних рішень.

- 1) Принцип лінгвістичності вхідних та вихідних змінних.

Відповідно до цього принципу, входи об'єкту та його вихід розглядається як лінгвістичні змінні, які оцінюються якісними термами.

Згідно Л. Заде, лінгвістичною змінною називається така змінна, значеннями якої є слова або речення природньої мови.

а. Принцип формування структури залежності «вхід вихід» в нечіткій базі знань.

Нечітка база знань являє собою сукупність правил ЯКЩО входи, ТО вихід, котрі відбивають досвід експерта і розуміння причинно-наслідкових зв'язків.

б. Принцип ієрархічності баз знань.

Використання цього принципу дозволяє пересилити «прокляття розмірності». За великої кількості вхідних перемінних побудова системи висловлювань про невідому залежність «входи-виходи» стає ускладненою. Це обумовлено тим, що в оперативній пам'яті людини може утримуватись не більше 7 понять-ознак. В зв'язку з цим необхідно провести класифікацію вхідних змінних і по ній побудувати дерево виводу, що визначає систему вкладених один в одного висловлювань-знань різної розмірності.

За рахунок принципу ієрархічності можливо враховувати практично необмежену кількість вхідних змінних, що впливають на оцінку вихідної змінної. При побудові дерева виводу необхідно слідкувати за тим щоб число аргументів (вхідних стрілок) в кожному вузлі дерева не перевищувало 7. Доцільність багаторівневого представлення експертних знань зумовлена не тільки природньою ієрархічністю об'єктів ідентифікації, а і необхідністю врахування нових змінних по мірі накопичення знань про об'єкт.

с. Принцип термометра в оцінці якісних змінних.

Суть цього принципу полягає в тому, що експертна оцінка деякої змінної реалізується шляхом зафарбовування частини шкали (рис 2.1), ліва та права границя якої відповідає найменшому та найбільшому значенню розглядаємої змінної.

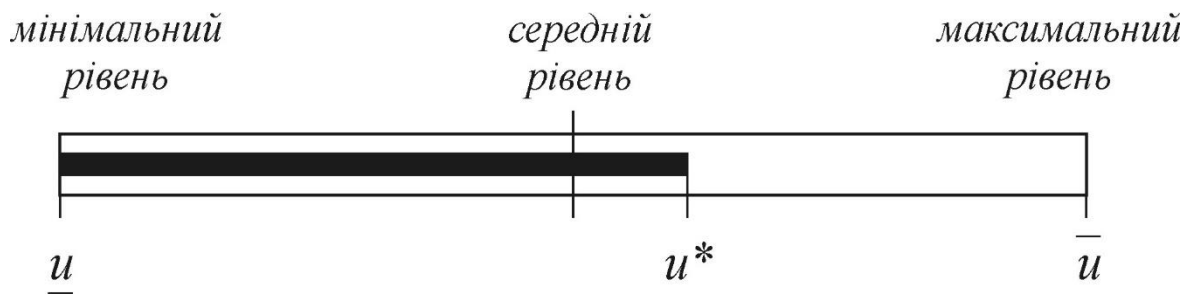


Рисунок 2.1 – оцінка змінної за принципом термометра

Принцип термометра зручно застосовувати у тих випадках, коли експерт не в змозі оцінити деяку змінну ні числом ні якісним термом, а лише інтуїтивно відчуває її рівень. Зручність такого підходу ще й в тому, що він дозволяє розглядати різні за своєю природою лінгвістичні змінні на єдиній універсальній множині.

d. Принцип двоетапного налаштування нечітких баз знань.

Згідно із цим принципом, побудова моделі нелінійного об'єкта

здійснюється в два етапи (рис 2.2), які за аналогією з класичними методами [8] можна вважати етапами структурної і параметричної ідентифікації. Параметрами, що налаштовуються, є вагомість нечітких правил ЯКЩО-ТО і форми функцій приналежності.

Перший етап є традиційним для технології нечітких експертних систем [4]. На ньому здійснюється формування і грубе налаштування моделі об'єкту шляхом побудови бази знань за доступною експертною інформацією.

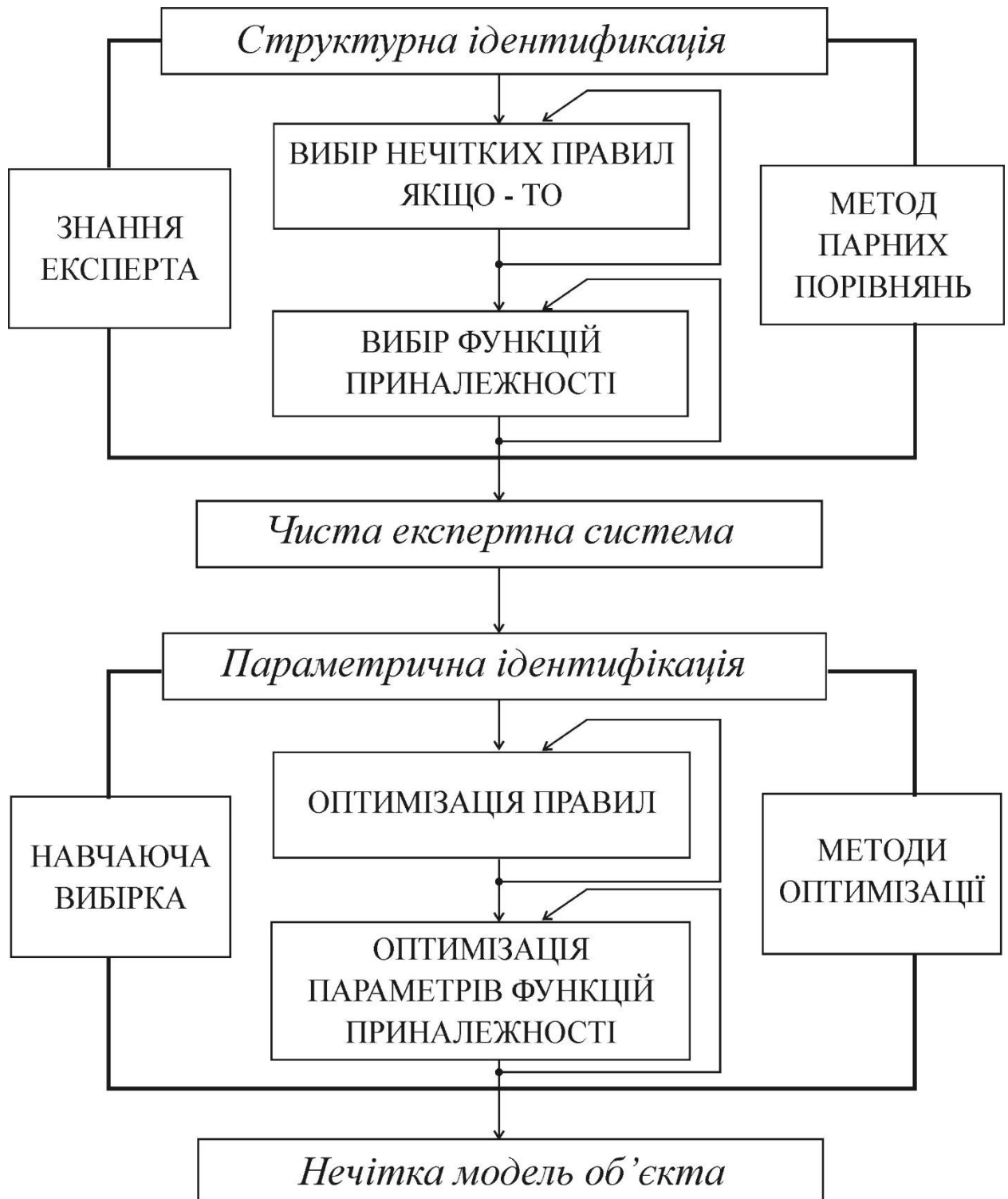


Рисунок 2.2 – Стапи налаштування нечіткої бази знань

Для грубого налаштування вагомості правил і форм функцій приналежності застосовується модифікований метод парних порівнянь Сааті, запропонований в [2]. Чим вищий професійний рівень експерта, тим вища

адекватність нечіткої моделі, побудованої на етапі грубого налаштування. Ця модель названа *чистою експертною системою*, оскільки для її побудови використовується лише експертна інформація. Однак, ніхто не може гарантувати збігу результатів нечіткого логічного висновку (теорія) та експериментальних даних. Тому необхідний другий етап, на якому здійснюється тонке налаштування нечіткої моделі шляхом її навчання за експериментальними даними.

Суть етапу тонкого налаштування складається у підборі таких вагів нечітких правил ЯКЩО-ТО і таких параметрів функцій приналежності, які мінімізують різницю між бажаною (експериментальною) і модельною (теоретичною) поведінкою об'єкта.

Етап тонкого налаштування формулюється як задача нелінійної оптимізації, яка може рішатися різними методами, серед яких найбільш універсальним є найшвидший спуск [2]. Однак, при великій кількості вхідних перемінних і нечітких термів у базі знань, застосування методу найшвидшого спуску вимагає пошуку мінімуму з різних початкових точок, що суттєво збільшує затрати машинного часу. Тому нами пропонується тонке налаштування нечіткої бази знань із застосуванням генетичних алгоритмів оптимізації [4]. Ці алгоритми є аналогом випадкового пошуку [4], який ведеться одночасно із різних початкових точок, що скорочує час пошуку оптимальних параметрів нечіткої моделі.

### **2.3 Постановка задачі моделювання оцінки якості растрових зображень**

Якість зображень і пристроїв характеризується величиною спотворень які вони вносять при відтворенні, наприклад, тестового зображення. Найважливішими при цьому є три види спотворень. Це спотворення кольору, шум, та геометричні спотворення. Для усунення суб'єктивної складової подібні тесування проводять на автоматизованих системах методом порівняння сфотографованих або відсканованих зображень спеціально розроблених та



стандартизованих тестів з їх електронними еталонами, так як це показано в [1,2]. Власне оцінювання відбувається з використанням ймовірностних та Байесовських моделей[3].

Практика використання систем оцінки якості, заснованих на ймовірностних моделях показала деяку їх недостатність і потребу врахування психо-фізіологічних особливостей людини У роботі[3] показано доцільність використання для подібних підходів методології експертних систем заснованих на базах знань, але при цьому ми маємо поєднання двох типів формалізації невизначеності: «ймовірно» та «можливо» що робить кінцеву модель достатньо громіздкою, а при взаємних перетвореннях призводить до втрати адекватності. Пропонується відмовитись від формалізації невизначеності через функцію розподілення теорії ймовірності і замінити її функцією належності із теорії нечітких множин.

Традиційно для оцінки якості скажімо цифрових фотоапаратів професіонали вживають лінгвістичні вирази, які беззаперечно можна віднести до термів нечіткої логіки, а самі множини показників, що відповідають тому чи іншому терму до нечітких множин. Так, наприклад, величину шумів цифрового фотознімка, як правило, характеризують слідуючими термами: «дуже малі», «малі», «помірні», «середні», «значні», «великі», «дуже великі». Теж саме стосується інших видів спотворень, таких наприклад як дисторсія (геометричні спотворення) та кольороспотворення.

#### **2.4 Моделювання оцінювання якості растрових зображень як ієрархічний об'єкту**

Трьохступеневий ієрархічний взаємозв'язок між інтегрованим показником якості  $R(\text{quality})$ , частинними показниками  $X(\text{noise})$ ,  $Y(\text{color})$ ,  $Z(\text{distortion})$ , та вхідними змінними доцільно представити у вигляді дерева логічного виведення, якому відповідає система співвідношень:

$$R=f_R(x,y,\dots,z) , \quad (2.1)$$

$$X=f_X(x_1,x_2, \dots, x_j) , \quad (2.2)$$

$$Y=f_Y(y_1,y_2, \dots, y_m) , \quad (2.3)$$

$$Z=f_Z(z_1,z_2, \dots, z_n) , \quad (2.4)$$

де  $R$  – вихідна змінна (інтегральний показник);

$X, Y, \dots, Z$  – класи вхідних змінних;

$x_i, y_j, z_k$  – вхідні змінні, віднесені до класів  $X, Y, \dots, Z$ ,

причому  $i=\overline{1, l}, j=\overline{1, m}, k=\overline{1, n}$ .

Дерево на рисунку 2.3 відображає тривірневу ієрархію вхідних змінних.

Вважатимемо, що всі змінні, які стоять у вершинах дерева (рис 3.3), є

лінгвістичними змінними з наступними термами:

$\{R_1, R_2, \dots, R_r\}$  – множина термів для оцінки змінної  $R$ ;

$\{X_1, X_2, \dots, X_a\}$  – множина термів для оцінки змінної  $X$ ;

$\{Y_1, Y_2, \dots, Y_b\}$  – множина термів для оцінки змінної  $Y$ ;

$\{Z_1, Z_2, \dots, Z_c\}$  – множина термів для оцінки змінної  $Z$ ;

$\{x_{i1}, x_{i2}, \dots, x_{ia_i}\}$  – множина термів для оцінки змінної  $x_i, i = \overline{1, l}$ ;

$\{y_{j1}, y_{j2}, \dots, y_{jb_j}\}$  – множина термів для оцінки змінної  $y_j, j = \overline{1, m}$ ;

$\{z_{k1}, z_{k2}, \dots, z_{kc_k}\}$  – множина термів для оцінки змінної  $z_k, k = \overline{1, n}$ .

Всередині кожної із множин нечіткі терми упорядкованості за принципом від нижчого до вищого, наприклад {низький, середній, високий тощо}.

Потужності всіх множин, тобто, числа термів, котрі використовуються для оцінки лінгвістичних змінних, які входять в співвідношення (2.1)-(2.4), в загальному випадку можуть відрізнятися.

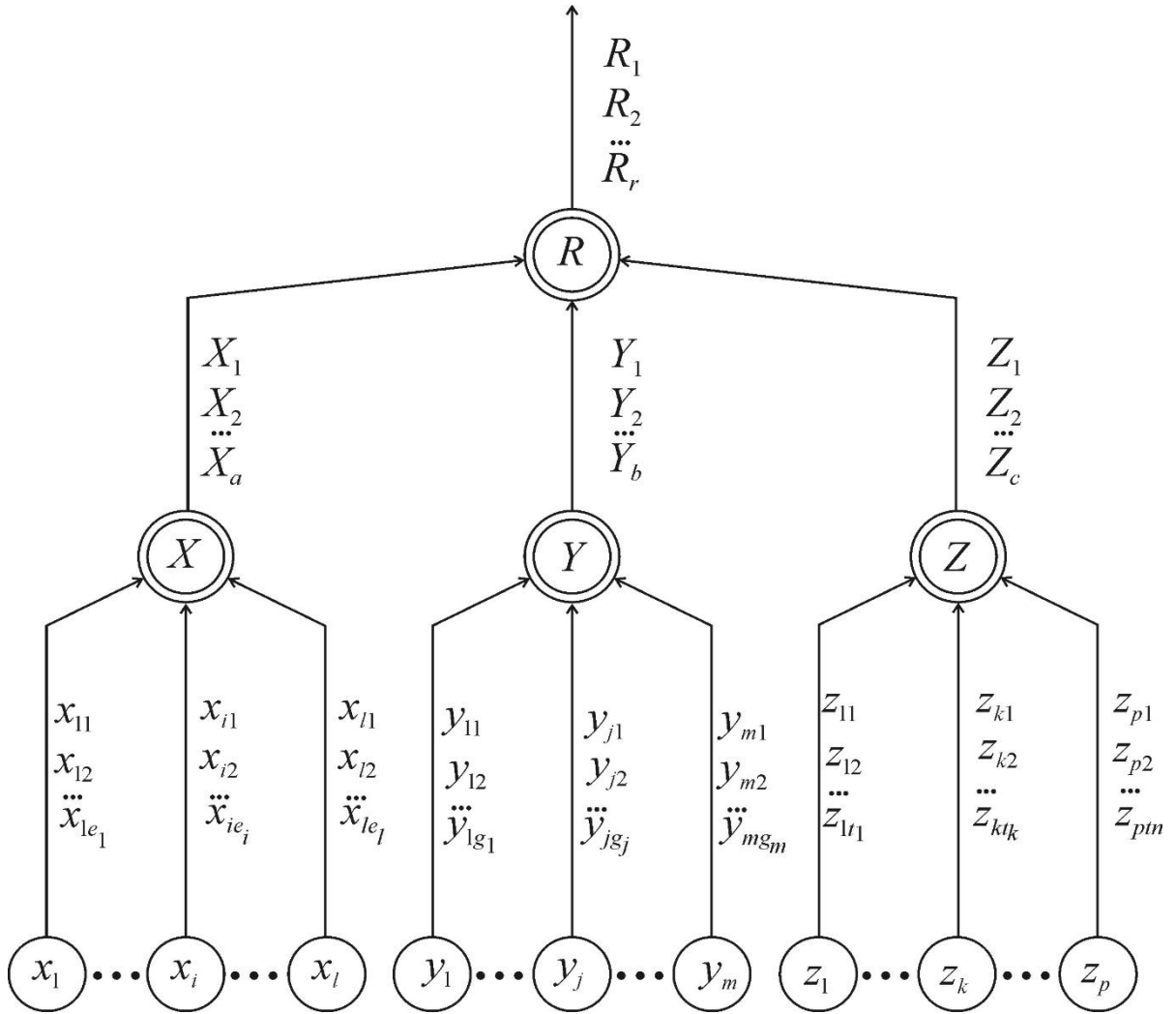


Рисунок 2.3 – Дерево логічного виведення інтегрованого показника якості

Використовуючи поняття універсальної множини і функції належності, кожен з термів представимо у вигляді нечіткої множини:

$$R_i = \int_W \mu^{R_i}(w)/w, i = \overline{1, r}, w \in W, \tag{2.5}$$

$$X_i = \int_{U_X} \mu^{X_i}(v_X)/v_X, i = \overline{1, a}, v_X \in U_X, \tag{2.6}$$

$$Y_i = \int_{U_Y} \mu^{Y_i}(v_Y)/v_Y, i = \overline{1, b}, v_Y \in U_Y, \tag{2.7}$$

$$Z_i = \int_{U_{z_i}} \mu^{z_i} (v_z)/v_z, \quad i = \overline{1, c}, \quad v_z \in U_{z_i}, \quad (2.8)$$

$$x_{ij} = \int_{U_{x_i}} \mu^{x_{ij}} (x_i)/x_i, \quad i = \overline{1, l}, \quad j = \overline{1, a_i}, \quad x_i \in U_{x_i}, \quad (2.9)$$

$$y_{jk} = \int_{U_{y_j}} \mu^{y_{jk}} (y_j)/y_j, \quad j = \overline{1, m}, \quad k = \overline{1, b_j}, \quad y_j \in U_{y_j}, \quad (2.10)$$

$$z_{kl} = \int_{U_{z_k}} \mu^{z_{kl}} (z_k)/z_k, \quad k = \overline{1, n}, \quad l = \overline{1, c_k}, \quad z_k \in U_{z_k}, \quad (2.11)$$

Де  $W$  – універсальна множина, на якій задана змінна  $R$ , тобто

$$R_i \subset W, \quad i = \overline{1, r};$$

$U_X, U_Y, U_Z$  – універсальні множини, на яких задано змінні  $X, Y, Z$ , тобто  $X_i \subset U_X$ ,

$Y_i \subset U_Y, Z_i \subset U_Z$ ;  $U_{x_i}, U_{y_j}, U_{z_k}$  – універсальні множини, на яких задані змінні

$x_i, y_j, z_k$ ,  $i = \overline{1, l}, j = \overline{1, m}, k = \overline{1, n}$ ;  $\mu^\xi(x)$  - функція приналежності змінної  $x$  нечіткому терму  $\xi$ .

Інформацію про співвідношення (2.1)-(2.4) будемо представляти у вигляді нечітких баз знань, які містять логічні висловлювання про взаємозв'язок вхідних та вихідних змінних [4].

Нечітка база знань про співвідношення (2.1) буде мати наступний вигляд:

ЯКЩО

$$(X = X^{j1}) \text{ та } (Y = Y^{j1}) \text{ та ... та } (Z_l = x^{j1}) \text{ або}$$

$$(X = X^{j2}) \text{ та } (Y = Y^{j2}) \text{ та ... та } (Z_l = x^{j2}) \text{ або}$$

$$(X = X^{jh_j}) \text{ та } (Y = Y^{jh_j}) \text{ та ... та } (Z_l = x^{jh_j}),$$

$$TO R = R_j, j = \overline{1, r}. \quad (2.12)$$

Ці правила можна представити у вигляді матриці знань  $M_R$

$$M_R = \begin{array}{|c|c|c|c|} \hline & X & Y & Z & R \\ \hline & X^{11} & Y^{11} & x^{11} & R_1 \\ \hline & \dots \\ & X^{1h_1} & Y^{1h_1} & x^{1h_1} & \\ \hline & x^{j1} & Y^{j1} & x^{j1} & \dots \\ & \dots \\ & x^{jh_j} & Y^{jh_j} & x^{jh_j} & R_j \\ & & & & \dots \\ \hline & x^{r1} & Y^{r1} & x^{r1} & R_r \\ & \dots \\ & x^{rh_r} & Y^{rh_r} & x^{rh_r} & \\ \hline \end{array} \quad (2.13)$$

Із застосуванням операцій перетину ( $\cap$ ) та об'єднання ( $\cup$ ) множин перепишемо нечітку базу знань (2.12) в наступній формі:

$$\cup_{p=1}^{h_j} [(X = X^{jp}) \cap (Y = Y^{jp}) \cap \dots \cap (Z = Z^{jp})] \rightarrow R_j, j = \overline{1, r}. \quad (2.14)$$

Нечітка база знань про співвідношення (2.2) буде мати наступний вигляд:

ЯКЩО

$$\begin{aligned} & (x_1 = x_1^{j1}) \text{ та } (x_2 = x_2^{j1}) \text{ та } \dots \text{ та } (x_l = x_l^{j1}) \text{ або} \\ & (x_1 = x_1^{j2}) \text{ та } (x_2 = x_2^{j2}) \text{ та } \dots \text{ та } (x_l = x_l^{j2}) \text{ або} \\ & (x_1 = x_1^{je_j}) \text{ та } (x_2 = x_2^{je_j}) \text{ та } \dots \text{ та } (x_l = x_l^{je_j}), \\ & \text{ТО } X = X_j, j = \overline{1, a}. \end{aligned} \quad (2.15)$$

Ці правила можна представити у вигляді матриці знань  $M_X$ :

$$M_X = \begin{array}{|c|c|c|c|} \hline x_1 & x_2 & \dots & X \\ \hline & & x_l & \\ \hline x_1^{11} & x_2^{11} & x_l^{11} & X_1 \\ \hline \dots & \dots & \dots & \\ \hline x_1^{1e_1} & x_2^{1e_1} & x_l^{1e_1} & \\ \hline x_1^{j1} & x_2^{j1} & x_l^{j1} & \dots \\ \hline \dots & \dots & \dots & X_j \\ \hline \dots & \dots & \dots & \dots \\ \hline x_1^{je_j} & x_2^{je_j} & x_l^{je_j} & \\ \hline x_1^{a1} & x_2^{a1} & x_l^{a1} & X_a \\ \hline \dots & \dots & \dots & \\ \hline x_1^{ae_a} & x_2^{ae_a} & x_l^{ae_a} & \\ \hline \end{array} \quad (2.16)$$

Із застосуванням операцій перетину ( $\cap$ ) та об'єднання ( $\cup$ ) множин перепишемо нечітку базу знань (2.15) в наступній формі:

$$\cup_{p=1}^{e_j} \left[ \cap_{i=1}^l (x_i = x_i^{jp}) \right] \rightarrow X = X_j, \quad j = \overline{1, a}. \quad (2.17)$$

Нечітка база знань про відповідність (2.3) буде мати наступний вигляд:

$$\begin{aligned} &\text{ЯКЩО } (y_1 = y_1^{j1}) \text{ та } (y_2 = y_2^{j1}) \text{ та } \dots \text{ та } (y_m = y_m^{j1}) \text{ АБО} \\ &(y_1 = y_1^{j2}) \text{ I } (y_2 = y_2^{j2}) \text{ I } \dots \text{ I } (y_m = y_m^{j2}) \text{ АБО } \dots \\ &(y_1 = y_1^{jq_j}) \text{ I } (y_2 = y_2^{jq_j}) \text{ I } \dots \text{ I } (y_m = y_m^{jq_j}), \\ &\text{ТО } Y = Y_j, \quad j = \overline{1, b}. \end{aligned} \quad (2.18)$$

Ці правила можна представити у вигляді матриці знань  $M_Y$ :

	y <sub>1</sub>	y <sub>2</sub>	... y <sub>m</sub>	Y
$M_Y =$	y <sub>1</sub> <sup>11</sup>	y <sub>2</sub> <sup>11</sup>	y <sub>m</sub> <sup>11</sup>	Y <sub>1</sub>
	...	...	...	
	y <sub>1</sub> <sup>1q<sub>1</sub></sup>	y <sub>2</sub> <sup>1q<sub>1</sub></sup>	y <sub>m</sub> <sup>1q<sub>1</sub></sup>	
	y <sub>1</sub> <sup>j1</sup>	y <sub>2</sub> <sup>j1</sup>	y <sub>m</sub> <sup>j1</sup>	Y <sub>j</sub>
	...	...	...	...
	y <sub>1</sub> <sup>jq<sub>j</sub></sup>	y <sub>2</sub> <sup>jq<sub>j</sub></sup>	y <sub>m</sub> <sup>jq<sub>j</sub></sup>	
	y <sub>1</sub> <sup>b1</sup>	y <sub>2</sub> <sup>b1</sup>	y <sub>m</sub> <sup>b1</sup>	Y <sub>b</sub>
	...	...	...	
	y <sub>1</sub> <sup>bq<sub>b</sub></sup>	y <sub>2</sub> <sup>bq<sub>b</sub></sup>	y <sub>m</sub> <sup>bq<sub>b</sub></sup>	

(2.19)

З використанням операцій перетину ( $\cap$ ) і об'єднання ( $\cup$ ) множин перепишемо нечітку базу знань (2.18) в наступній формі:

$$\cup_{p=1}^{q_j} [\cap_{i=1}^m (y_i = y_i^{jp})] \rightarrow Y = y_j, j = \overline{1, b}. \quad (2.20)$$

Нечітка база знань про відповідність (2.4) буде мати наступний вигляд:

ЯКЩО  $(z_1 = z_1^{j1}) \text{ I } (z_2 = z_2^{j1}) \text{ I } \dots \text{ I } (z_n = z_n^{j1})$  АБО  
 $(z_1 = z_1^{j2}) \text{ I } (z_2 = z_2^{j2}) \text{ I } \dots \text{ I } (z_n = z_n^{j2})$  АБО...  
 $(z_1 = z_1^{jt_j}) \text{ I } (z_2 = z_2^{jt_j}) \text{ I } \dots \text{ I } (z_n = z_n^{jt_j})$ ,  
 ТО  $Z = Z_j, j = \overline{1, c}$  (2.21)

Ці правила можна представити у вигляді матриці знань  $M_Z$ :

$$M_Z = \begin{array}{|c|c|c|c|} \hline & z_1 & z_2 & \dots \\ \hline & & & z_n & Z \\ \hline z_1^{11} & z_2^{11} & z_n^{11} & Z_1 \\ \dots & \dots & \dots & \\ z_1^{1t_1} & z_2^{1t_1} & z_n^{1t_1} & \\ \hline z_1^{j1} & z_2^{j1} & z_n^{j1} & \dots \\ \dots & \dots & \dots & Z_j \\ \dots & \dots & \dots & \dots \\ z_1^{jt_j} & z_2^{jt_j} & z_n^{jt_j} & \\ \hline z_1^{c1} & z_2^{c1} & z_n^{c1} & Z_c \\ \dots & \dots & \dots & \\ z_1^{ct_c} & z_2^{ct_c} & z_n^{ct_c} & \\ \hline \end{array} \quad (2.22)$$

З використанням операцій перетину ( $\cap$ ) і об'єднання ( $\cup$ ) множин перепишемо нечітку базу знань (2.21) в наступній формі:

$$\cup_{p=1}^{t_j} [\cap_{i=1}^n (z_i = z_i^{jp})] \rightarrow Z = z_j, j = \overline{1, c}. \quad (2.23)$$

Таким чином нами визначена система нечітких логічних висловлювань (2.14), (2.17), (2.20), (2.23), які описують експертну інформацію про співвідношення (2.1)-(2.4), що відповідає узагальненому дереву взаємозв'язку «вхідні змінні – вихідна змінна». Принципи кодування лінгвістичних змінних і їх нечітких термів, запропоновані вище для дворівневого дерева виведення, залишаються незмінними для дерева довільної розмірності. Структура дерева, вибір термів лінгвістичних змінних на його вітках, а також нечіткі бази знань в кожній з нетермінальних вершин дерева, - все це визначає методику збору експертної інформації, необхідної для побудови моделі нелінійного об'єкту.



## 2.5 Представлення баз знань нечіткими логічними рівняннями

Поставимо у відповідність співвідношенням (2.1)-(2.4) нечіткі логічні рівняння, що зв'язують функції належності нечітких термів вхідних і вихідних змінних. Для отримання цих рівнянь замінимо нечіткі множини в лівих і правих частинах висловлювання (2.14), (2.17), (2.20) та (2.23) відповідними функціями приналежності, а операції І та АБО над нечіткими множинами замінимо операціями  $\wedge(\min)$  та  $\vee(\max)$  над відповідними функціями приналежності.

Із виразу (2.14) отримаємо:

$$\mu^{R_j}(X, Y, Z) = \bigvee_{p=1}^{h_j} \left\{ \min \left[ \mu^{X^{jp}}(X) \wedge \mu^{Y^{jp}}(Y) \wedge \mu^{Z^{jp}}(Z) \right] \right\}, \quad (2.24)$$

де  $\mu^{R_j}(X, Y, Z)$  - поверхня приналежності вектора змінних  $X, Y, Z$  нечіткому терму-оцінці  $R_j$ ;

$\mu^{X^{jp}}(X), \mu^{Y^{jp}}(Y), \mu^{Z^{jp}}(Z)$  – функції приналежності змінних  $X, Y, Z$  нечітким термам  $X^{jp}, Y^{jp}, Z^{jp}$ , відповідно  $p = \overline{1, h_j}, j = \overline{1, r}$ .

З виразу (2.17) отримаємо:

$$\mu^{X_j}(x_1, x_2, \dots, x_l) = \bigvee_{p=1}^{e_j} \left[ \bigwedge_{i=1}^l \mu^{x_i^{jp}}(x_i) \right], \quad (2.25)$$

де  $\mu^{X_j}(x_1, x_2, \dots, x_l)$  - поверхня приналежності вектора змінних  $(x_1, x_2, \dots, x_l)$  нечіткому терму-оцінці  $X_j$ ;

$\mu_i^{jp}(x_i)$  – функція приналежності змінної  $x_i, i = \overline{1, l}$ , нечіткому терму  $x_i^{jp}$ ,  $p = \overline{1, e_j}, j = \overline{1, a}$ .

З виразу (2.20) отримуємо:

$$\mu^{Y_j}(y_1, y_2, \dots, y_m) = \bigvee_{p=1}^{g_j} \left[ \bigwedge_{i=1}^m \mu^{Y_i^{jp}}(y_i) \right], \quad (2.26)$$

де  $\mu^{Y_j}(y_1, y_2, \dots, y_m)$  - поверхня приналежності вектора змінних  $(y_1, y_2, \dots, y_m)$  нечіткому терму –оцінці.

$\mu^{Y_i^{jp}}(y_i)$  - функція приналежності змінної  $y_i$ ,  $i = \overline{1, m}$  нечіткому терму  $\mu^{Y_i^{jp}}$ ,  $p = \overline{1, g_j}$ ,  $j = \overline{1, b}$ .

З виразу (2.23) отримаємо:

$$\mu^{Z_j}(z_1, z_2, \dots, z_n) = \bigvee_{p=1}^{t_j} \left[ \bigwedge_{i=1}^n \mu^{z_i^{jp}}(z_i) \right], \quad (2.27)$$

де  $\mu^{Z_j}(z_1, z_2, \dots, z_n)$  - поверхня приналежності вектора змінних  $(z_1, z_2, \dots, z_n)$  нечіткому терму-оцінці  $Z_j$ ;

$\mu^{z_i^{jp}}(z_i)$  – функція приналежності змінної  $z_i$ ,  $i = \overline{1, n}$  нечіткому терму  $z_i^{jp}$ ,  $p = \overline{1, t_j}$ ,  $j = \overline{1, c}$ .

Враховуючи те, що операціями  $\bigwedge$  та  $\bigvee$  в теорії нечітких множин відповідають операції *min* та *max*, перепишемо співвідношення (2.24)-(2.27) у вигляді:

$$\mu^{R_j}(X, Y, Z) = \max_{p=1, h_j} \left\{ \min \left[ \mu^{X^{jp}}(X), \mu^{Y^{jp}}(Y), \mu^{Z^{jp}}(Z) \right] \right\}, \quad (2.28)$$

$$\mu^{X_j}(x_1, x_2, \dots, x_l) = \max_{p=1, e_j} \left\{ \min_{i=1, l} \left[ \mu^{X_i^{jp}}(X_i) \right] \right\}, \quad (2.29)$$

$$\mu^{Y_j}(y_1, y_2, \dots, y_m) = \max_{p=1, g_j} \left\{ \min_{i=1, m} \left[ \mu^{Y_i^{jp}}(Y_i) \right] \right\}, \quad (2.30)$$

$$\mu^{Z_j}(z_1, z_2, \dots, z_j) = p = \frac{\max}{1, t_j} \left\{ \frac{\min}{1, n} \left[ \mu^{Z_i^{j_p}}(Z_i) \right] \right\}, \quad (2.31)$$

## 2.6 Розробка алгоритму виведення узагальненої оцінки

У попередньому підрозділі нами отримана система співвідношень (2.24)-(2.27), яка повністю відповідає узагальненому дереву логічного виведення (рис 2.3) і дозволяє вирахувати ступені приналежності вектора значень вхідних змінних нечітким термам-оцінкам вихідної змінної. Алгоритм нечіткого логічного виведення, який використовує узагальнене дерево виведення має вигляд:

1. Зафіксуємо вектор значень вхідних змінних  $(x_1^*, x_2^*, \dots, x_l^*, y_1^*, y_2^*, \dots, y_m^*, z_1^*, z_2^*, \dots, z_n^*)$ .
2. Визначимо значення функцій належності термів-оцінок вхідних змінних  $\mu^{x_{ij}}(x_i), i = \overline{1, l}, j = \overline{1, e_l}, \mu^{y_{jk}}(y_j), j = \overline{1, m}, k = \overline{1, q_j}, \mu^{z_{kl}}(z_k), k = \overline{1, n}, l = \overline{1, t_k}$ .
3. Використовуючи співвідношення (2.56)-(2.59) вирахуємо функції приналежності  $\mu^{R_j}(X, Y, Z)$  термів-оцінок вихідної величини R, яка відповідає вектору значень вхідних змінних  $(x_1^*, x_2^*, \dots, x_l^*, y_1^*, y_2^*, \dots, y_m^*, z_1^*, z_2^*, \dots, z_n^*)$ .
4. Визначимо оцінку  $R_j^*$ , функція приналежності якої максимальна:

$$\mu^{R_j^*}(X, Y, Z) = \max[\mu^{R_j}(X, Y, Z)] \rightarrow R = R_j^*, j = \overline{1, r} \quad (2.32)$$

Послідовність розрахунків, які відповідають алгоритму, приведена на рис 2.4.

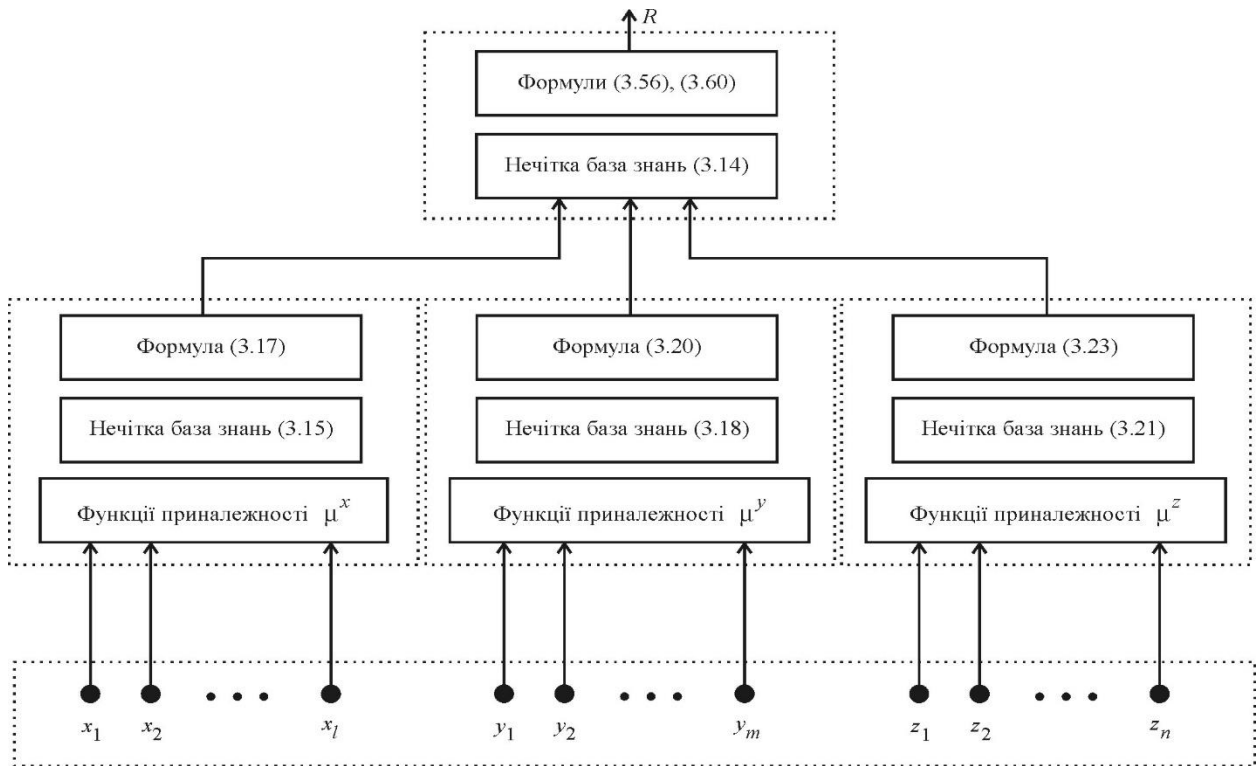


Рисунок 2.4 - Послідовність розрахунків нечіткого логічного виведення

## 2.7 Формалізація моделі узагальненого логічного виводу оцінки якості фотокамери

Для формалізації моделі узагальненого логічного виводу оцінки фотокамери скористаємося апаратом нечіткої логіки. Ієрархічний взаємозв'язок між вхідними змінними, що описуються нечіткими термами типу «незначні», «малі», «середні», «великі», «дуже великі», що стосується, наприклад геометричних спотворень, або «відповідає», «не відповідає», що стосується роздільної здатності, класів вхідних змінних і вихідною змінною (узагальненою оцінкою фотокамери) представимо у вигляді дерева (Рис.3.5), якому відповідає система співвідношень

$$R = \int_R (X, Y, Z), \quad (2.33)$$

$$X = \int_X (D, C), \quad (2.34)$$

$$Y = \int_Y (E, K, ), \quad (2.35)$$

$$Z = \int_Z (L, B), \quad (2.36)$$

де  $R$ - вихідна змінна (інтегрований узагальнений показник);

$X, Y, Z$  – класи вхідних змінних;

$A, B, C, D, E, K, L$  – підкласи вхідних змінних;

$a_i, b_i, c_i, d_i, e_i, k_i, l_i$  - вхідні нечіткі змінні, що відносяться до відповідних підкласів.

Будемо вважати, що всі змінні, що стоять у вершинах дерева (Рис.2.5), є лінгвістичними змінними зі слідуючими нечіткими термами:

{2-, 2, 2+, 3-, 3, 3+, 4-, 4, 4+, 5-, 5, 5+} – множина термів для оцінки змінної  $R$ ;

{«незначні», «малі», «середні», «великі», «дуже великі»} – множина термів для оцінки змінних  $X$ (геометричні спотворення),  $D$ (дисторсія),  $Y, K$ (кома),

$A, B, Z$ (хроматична аберація каналу «а», «в» та загальне).

{«відповідає», «не відповідає»} - множина термів для оцінки змінної  $C$ (роздільна здатність),

{«яскраві», «не яскраві»} - множина термів для оцінки змінної  $L$ (яскравість),

{«сприятливі», «не сприятливі»} - множина термів для оцінки змінної  $E$ (умови зйомки).

На рисунку 2.5 показано графічну модель узагального виведення оцінки.

## 2.8 Розробка діаграми класів програмного забезпечення системи

Для реалізації цієї моделі розроблено структуру програмного забезпечення системи оцінки якості цифрових фотоапаратів в концепції UML. Система самостійно виконує тестування цифрових фотоапаратів на кольороспотворення, геометричні спотворення, граничну роздільну здатність і шуми в повному діапазоні ISO. При цьому вона автоматично регулює рівень освітлення та вибирає режими фотокамери. Тестування проводиться в кольорових моделях Lab або CMYK. Система визначає інтегровану оцінку якості оцифрованого фотокамерою зображення за 12-бальною шкалою з врахуванням ринкових трендів і формує власну базу даних протестованих фотокамер.

На рис. 2.6 показано структуру програмного забезпечення системи, що орієнтована на реалізацію будь якою об'єктно-орієнтованою мовою програмування. В основі роботи системи лежить принцип порівняння введеного оцифрованого зображення з програмно синтезованим ідеальним еталоном.

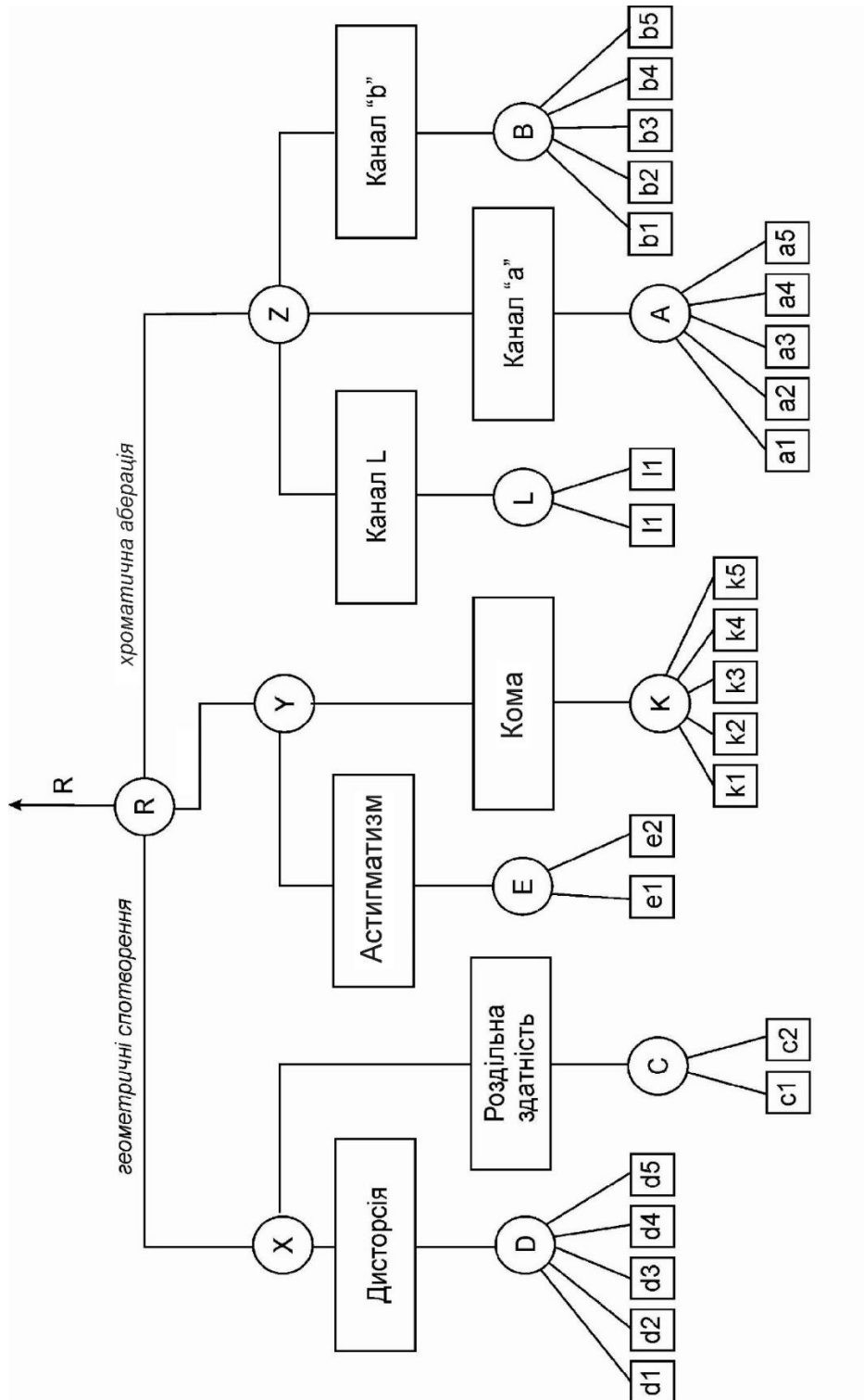


Рисунок 2.5 – Дерево виводу інтегрального показника геометричного спотворення

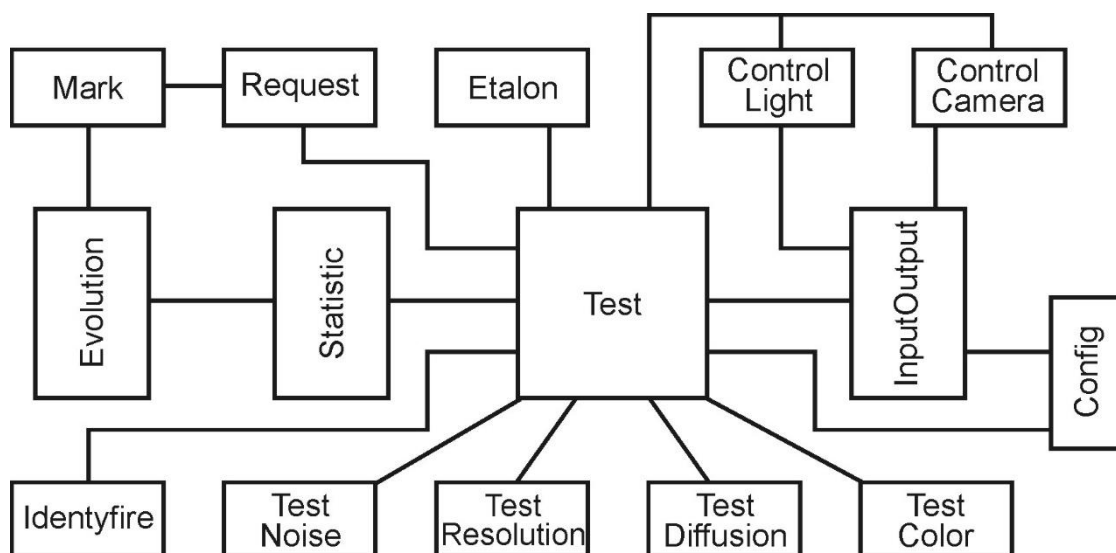


Рисунок 2.6 - Діаграма класів ПЗ-системи

Визначення рівня шумів та спотворень виконується класом TEST з агрегованими класами TestColor, TestNoise, TestResolution, TestDiffusion та класом Etalon, де зберігаються синтезовані цифрові еталони. Тут же формуються повідомлення класам ControlLight та ControlCamera для зміни режимів освітлення та роботи фотокамери. Власне оцінювання відбувається в класі Identyfire з використанням ймовірностних та Байєсовських моделей. Разом із вводом оцифрованого зображення вводиться паспорт цифрової фотокамери, який разом із результатами тестування передається класу Statistica, де формується власна база даних проведених досліджень.

Клас Request формує інтегровану оцінку у 12-бальній системі, а клас Mark коректує одержану оцінку з врахуванням еволюційних та риночних трендів, за формування яких відповідає клас Evolution

Для достовірної роботи класів Mark, Evolution, Statistic система повинна пройти деякий період навчання. Система може бути адаптована для тестування сканерів та цифрових відеокамер.

## 2.9 Математична модель тестування на шумові завади

Метод виявлення шумів в зображенні базується на фотографуванні тестової картинки (рис.2.7), яка представляє собою головні кольори кожного з каналів кольорової моделі ( RGB, CMYK, Lab ).



Рис. 2.7 Тест картинка

Кожен з кольорів моделі (С, М, Y, К) представляє собою масив  $\bar{Y}_\phi$ , який складається з елементів  $y_{ij}^k$ :

$$y_{ij}^k \in \bar{Y}_\phi \quad (2.37)$$

Припустимо, що критерієм шуму є дисперсія  $\sigma^2\{y_{ij}^k\}$ . Так як дисперсія є двосторонньою, тому використовувати її в якості критерію для оцінки шуму незручно. В даному випадку, в якості критерію шуму, краще використовувати математичне очікування відхилень від  $y_{вим}^k$ .

Максимальним числом системи кодування кольору у відсотковому відношенні вважається 100%. При фотографуванні отримуємо по 50% кожного з кольорів.

Максимальний рівень шуму для певного класу фотоапаратів  $y_{\max}^k$  береться з експертних оцінок. Пороговими значеннями шуму при певних параметрах



чутливості (ISO 80, 160 та 320) є 5%, 10% та 20% відповідно. Всі отримані значення які нижчі порогових не вважаються шумом. Якщо в результаті отримані дані перевищують порогові значення, то необхідно знайти їх математичне очікування  $y_{вим}^k$  :

$$y_{ij}^k - y_{вим}^k \geq y_{пор}^k \quad (2.38)$$

де  $y_{пор}^k = y_{\max}^k / 20$

Отже, математичне очікування  $y^k = E\{\bar{Y}_{ш}\}$  є критерієм оцінки шуму.

За допомогою функції відповідності (3.2) визначається оцінка яку отримує певний цифровий фотоапарат при тестуванні на шуми:

$$X = \frac{y_{\max}^k}{y^k} - 1 \quad (2.39)$$

## 2.10 Визначення рівня шумів на основі шумів медіанної фільтрації

Тестування рівня шумів проводитиметься на основі медіанної фільтрації зображень. Медіанний фільтр являє собою ковзне вікно, яке зазвичай охоплює непарну кількість вибірок  $N$  аналогового сигналу  $x(t)$ . Вихідною величиною фільтра  $y_j$  є відрахунок, для якого у вікні існує  $\frac{(N-1)}{2}$  відрахунків менших або рівних йому за величиною та стільки ж більших або рівних йому за величиною:

$$y_j = \text{med}\{x_0, x_1, \dots, x_j, \dots, x_{N-2}, x_{N-1}\} \quad (2.40)$$

Порядок фільтру  $N$  визначається розміром апертури (вікна), яке використовується для фільтрації. Найпростішим, за визначенням, є одновимірний медіанний фільтр із тривідрахунковим вікном. Для цього фільтра

сформульовані основні принципи апаратної реалізації, але для фільтрів вищих порядків вона виявляється дуже складною. Програмна реалізація виявляється суттєво простішою, оскільки для цифрових біполярних сигналів значення вихідного сигналу дорівнює арифметичній сумі

$$y_j = x_j + x_{j+1} + \dots + x_{j+N-2} + x_{j+N-1} \quad . \quad (2.41)$$

Медіанний фільтр характеризується нелінійним перетворенням сигналів, оскільки його властивості не передбачають виконання умови адитивності:

$$\begin{cases} med(k \cdot x(i)) = k \cdot med(x(i)), \\ med(a + x(i)) = a + med(x(i)), \\ med(x(i) \cdot g(i)) \neq med(x(i)) + med(g(i)), \end{cases} \quad (2.42)$$

де  $med(x)$  - оператор взяття медіани;  $k$ ,  $a$  - постійні;  $x(i)$ ,  $g(i)$  - послідовності вибірок довжиною  $N$ .

Оскільки реєстрацію значень, що надходять з каналу зв'язку, доцільно здійснювати з використанням АЦП, то необхідно визначити його основні параметри.

Мінімальна кількість відрахунків, які необхідно зафіксувати для очищення інформативного сигналу від шуму медіанним фільтром, визначається як  $N - \frac{1}{2} \cdot k_m$ , де  $k_m \geq 1$  - коефіцієнт запасу дискретизації сигналу.

$$T_{ADC} \leq \frac{2 \cdot k_m}{(N-1) \cdot v \cdot k_v} - T_{WR} - T_{RD}, \quad (2.43)$$

де  $T_{ADC}$  тривалість циклу перетворення АЦП;  $T_{WR}$  - тривалість програмного циклу запуску АЦП за допомогою інтерфейсної схеми до моменту подання сигналу "Пуск";  $T_{RD}$  - тривалість програмного циклу зчитування даних з АЦП від моменту визначення сигналу "Кінець перетворення" до моменту записування даних до пам'яті.

Опорна напруга АЦП не повинна бути меншою, ніж вхідний сигнал  $x(t)$  з урахуванням похибки квантування. Для оцінювальних розрахунків можна скористатись спрощеною формулою:

$$U_{0.ADC} \geq \Delta U_{ADC} + v \cdot |U_c| + U_{\xi}, \quad (2.44)$$

$$U_{0.ADC} \cdot \left(1 - \frac{1}{N_{ADC}}\right) \geq |U_c| \cdot \left(v + \frac{1}{\sqrt{h^2}}\right). \quad (2.45)$$

Для двійкового аналого-цифрового перетворювача кінцева формула для визначення кількості розрядів АЦП незалежно від виду сигналу

$$n_{ADC} \geq \log_2 \left( \frac{U_{0.ADC}}{U_{0.ADC} - |U_c| \cdot \left(v + \frac{1}{\sqrt{h^2}}\right)} \right) + 1. \quad (2.46)$$

Задіяння коефіцієнту запасу медіанного фільтра надає нові властивості алгоритму фільтрації. Це дозволяє виключити ситуації, коли вхідний сигнал фільтра стає кореневим, тобто не змінюється під час проходження крізь нього. Крім цього, алгоритм медіанної фільтрації, побудований для рекурсивного режиму, має суттєво більшу спроможність для згладжування низькоамплітудного шуму.

Таким чином, протягом часу  $[0, T]$  на вхід медіанного фільтра поступає сукупність сигналів  $x(t)$ . Інформативний сигнал  $x(t - \tau)$  має невідоме часове розташування  $\tau \in [0, T]$ , причому на цей інтервал припадає багато елементів розділу за затримкою. Потік імпульсів  $\chi(t)$  має вигляд:

$$\chi(y) = \sum_{j=0}^{L-1} \kappa_{\chi \cdot j} \cdot U_{\chi \cdot j} \cdot f(t - \tau_{\chi \cdot j}), \quad (2.47)$$

де  $U_{\chi,j}$  – амплітуда імпульсу в потоці  $\chi(t)$ ;  $\tau_j$  – його часове розташування;  $\kappa_{\chi,j}$  – коефіцієнт наявності імпульсної завади, який дорівнює одиниці з імовірністю  $p_\chi$  та нулю – з імовірністю  $(1 - p_\chi)$ .

Таке завдання завади відповідає потоку Бернуллі, для якого на інтервалі  $[0, T]$  існує не більше  $L$  точок. Статистика кожної точки характеризується частковою щільністю

$$s_j(\tau_\chi) = p_{\chi,j} \cdot w_j(\tau_\chi), \quad (2.48)$$

де  $p_{\chi,j}$  – імовірність появи  $j$ -того імпульсу;  $w_j(\tau_\chi)$  – розподіл моментів їх появи. За умови виконання умови нормування  $\int_0^T w_j(\tau_\chi) d\tau_\chi = 1$ , при  $p_\chi=1$  (на інтервалі часу  $[0, T]$  наявні всі  $L$  імпульсів) та  $w_j(\tau_\chi) = \delta(\tau - \tau_j)$  потік  $\chi(t)$  визначається як детермінована імпульсна завада.

Якщо комбінований сигнал  $x(t)$  дискретизується за часом з інтервалом  $\Delta T$  і ці підрахунки піддаються ковзній рекурсивній медіанній фільтрації з апертурою  $N$ , то з урахуванням утворювальної функції  $\theta(z)$  потоку Бернуллі

$$\theta(z) = \prod_{i=0}^{N-1} (1 + p_{\chi,i} \cdot (z - 1)) = \sum_{i=0}^N p_{\chi,i} \cdot z^i, \quad (2.49)$$

де  $p_{\chi,i} = \frac{1}{i!} \cdot \frac{d\theta(z)}{dz^i}$ .

Можна записати імовірність вилучення імпульсної завади як:

$$p_{np} = \sum_{i=0}^{\frac{N-1}{2}} p_i, \quad (2.50)$$

або у випадку рівності всіх  $p_j$ , для потоку Бернуллі

$$p_{np}^{(B)} = \sum_{i=0}^{\frac{N-1}{2}} C_N^i \cdot p_\chi^i \cdot (1 - p_\chi)^{N-i}. \quad (2.51)$$

Якщо  $\frac{p_j}{\sum_{j=0}^{L-1} p_j} \ll 1$  то потік за своїми властивостями наближається до потоку

Пуассона, і

$$\Theta(z) = e^{\Lambda(z-1)}, \quad (2.52)$$

$$\text{де } \Lambda = \sum_{j=0}^{N-1} p_j \int_0^T w_j(\tau_\chi) d\tau_\chi = \sum_{j=0}^{N-1} p_j = \lambda \cdot N \quad ; \quad \lambda = \frac{1}{N} \sum_{j=1}^{N-1} p_j \quad -$$

середня інтенсивність пуасонівського потоку в межах апертури ковзного рекурсивного медіанного фільтра.

Значення  $N \cdot \Delta T$  характеризує часовий інтервал, на якому беруться  $N$  відрахунків. Тоді

$$p_{np}^{(P)} = \sum_{i=0}^{\frac{N-1}{2}} \frac{\Lambda^i}{i!} \cdot e^{-\Lambda}, \quad (2.53)$$

Результати розрахунків імовірностей вилучення імпульсної завади від імовірності  $p_\chi$  для рекурсивних медіанних фільтрів з різними апертурами наведені на рис. 2.7. Отримані результати показують високу ефективність використання фільтрів такого типу.

Для медіанного фільтра можна достатньо просто реалізувати алгоритм швидкого оброблювання, який базується на побудові різницевих матриць за допомогою порогової функції насичення  $F_{ij} = f(x_i - x_j)$ , в якій

$$f(\Delta x) = \begin{cases} 1, & \Delta x \geq 0 \\ 0, & \Delta x < 0 \end{cases} \quad (2.54)$$

Зсув на одну позицію вздовж ряду значень дає матрицю  $F1$ , в якій потрібно розраховувати лише дев'ять значень, розташованих у виділеній області.

$$\mathbf{F1} = \begin{pmatrix} F_{11} & F_{21} & F_{31} & F_{41} & \vdots & F_{51} \\ F_{12} & F_{22} & F_{32} & F_{42} & \vdots & F_{52} \\ F_{13} & F_{23} & F_{33} & F_{43} & \vdots & F_{53} \\ F_{14} & F_{24} & F_{34} & F_{44} & \vdots & F_{54} \\ \dots & \dots & \dots & \dots & \dots & \\ F_{15} & F_{25} & F_{35} & F_{45} & & F_{55} \end{pmatrix}, \quad (2.55)$$

$$F_i = \sum_{j=0}^N F_{ij} = \sum_{j=0}^N f(x_i - x_j). \quad (2.56)$$

Сума різниць значень  $F_{ij}$  за стовпчиками показує номер значення по величині  $i$  дає можливість сортувати зареєстровані значення  $x_j$  за величиною: 1 відповідає мінімальному,  $N$  – максимальному, а  $\frac{(N+1)}{2}$  – медіанному значенню.

Система оцінки шумів базується на визначенні співвідношення сигнал\шум, яке дає змогу визначити зашумленість зображення, та дає змогу надати кількісну оцінку якості зображення. Яке розраховується за допомогою відношення середнього значення сигналу до відхилення.

$$\text{ССШ} = \frac{P_{\text{сигн.}}}{P_{\text{шум}}} \quad (2.57)$$

## 2.11 Оцінювання рівня шумів методом нечіткої логіки

Визначення інтегрального та частинних показників оцінки шумів в цифрових фотоапаратах.

1. Позначимо інтегральний показник рівня шумів через  $R$ . Цей показник оцінюється за допомогою частинних показників:

$X$  – час проведення знімку, де  $x_1$  – зйомка проводиться вдень,  $x_2$  – зйомка проводиться вночі;

$Y$  – чутливість матриці, де  $y_1$  – 100 ISO,  $y_2$  – 200 ISO,  $y_3$  – 400 ISO,  $y_4$  – 600 ISO,  $y_5$  – 800 ISO.

2. Представимо схему ієрархії прийнятих показників у вигляді дерева виводу (рис. 2.8).

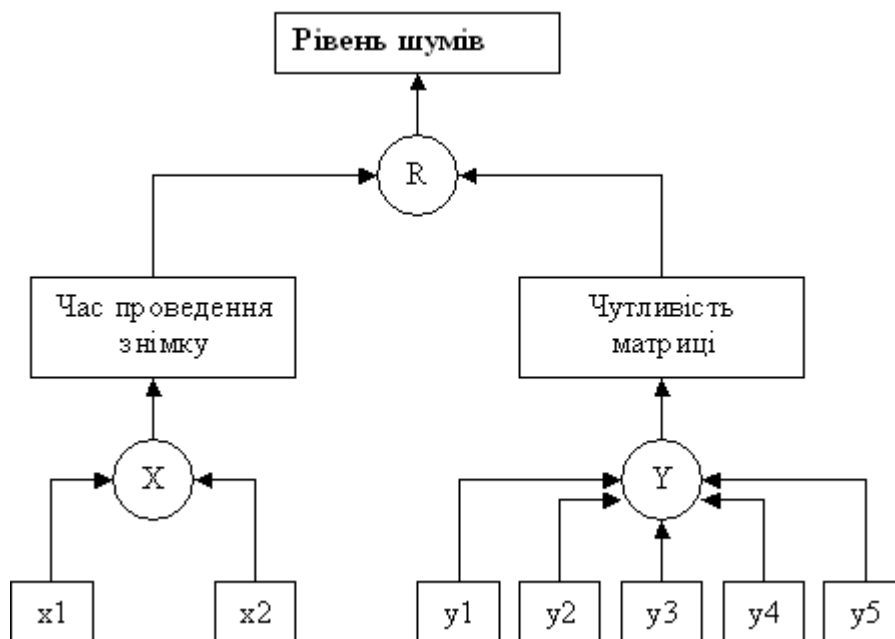


Рисунок 2.8 - Взаємозв'язок інтегрального та частинних показників

Дану схему представимо у вигляді співвідношень:

$$R = f_R ( X, Y ), \quad (2.58)$$

$$X = f_X ( x_1, x_2 ), \quad (2.59)$$

$$Y = f_Y ( y_1, \dots, y_5 ). \quad (2.60)$$

Цим співвідношенням надалі будуть поставлені у відповідність нечіткі логічні рівняння, які дозволяють обчислити рівень показника  $R$  за максимумом функції належності.

## 2.12 Модель оцінки інтегрального показника

1. Функції належності нечітких множин будують за допомогою універсальної шкали, на якій предметна шкала  $\bar{R}$  представлена за допомогою нечітких змінних  $\alpha_i$ , що відповідають термам базової терм-множини. Тут інтегральний показник  $R$  оцінюється нечіткими термами:

$\alpha_1$  – дуже малі (ДМ);

$\alpha_2$  – малі (М);

$\alpha_3$  – середні (С);

$\alpha_4$  – великі (В);

$\alpha_5$  – дуже великі (ДВ).

Ці терми описуються наступними значеннями:

ДМ = (1/0; 1/0,1; 0,5/0,15; 0/0,2);

М = (0/0; 0,5/0,15; 1/0,2; 1/0,3; 0,5/0,35; 0/0,4);

С = (0/0,3; 0,5/0,35; 1/0,4; 1/0,5; 1/0,6; 0,5/0,65; 0/0,7);

В = (0/0,6; 0,5/0,65; 1/0,7; 1/0,8; 0,5/0,85; 0/0,9);

ДВ = (0/0,8; 0,5/0,85; 1/0,9; 1/1).

2. Універсальна шкала поділяється на шкали  $\bar{X}$  та  $\bar{Y}$ . Шкалі  $\bar{X}$  відповідає частинний показник  $X$ , де лінгвістичні змінні  $x_1$  та  $x_2$  розподілені на інтервалах  $[0 \div 0,5]$  та  $[0,5 \div 1]$  відповідно. Шкалі  $\bar{Y}$  відповідає частинний показник  $Y$  на якій значення  $y_1, \dots, y_5$  розподілені на інтервалі  $[0 \div 800]$ .

3. Будуються функції відображення  $\pi_1$  та  $\pi_2$  на універсальній шкалі. Ці функції мають відповідні значення:

$\pi_1 = (0/0; 0,1/200; 0,2/400; 0,45/600; 1/800)$

$\pi_2 = (0/0; 0,4/200; 0,75/400; 0,9/600; 1/800)$



Функція відображення  $\pi_1$  відповідає частинному показнику  $X$  на шкалі  $\bar{X}$  що знаходиться в інтервалі  $[0 \div 0,5]$ , відповідно  $\pi_2$  відповідає частинному показнику  $X$  в інтервалі  $[0,5 \div 1]$ , (рис. 2.9).

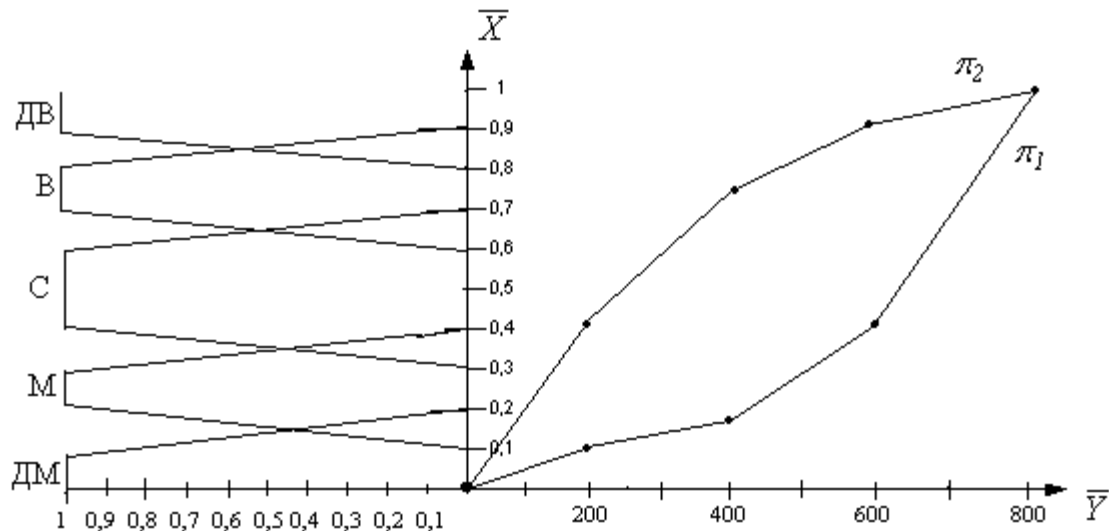


Рисунок 2.9 - Універсальна шкала оцінки інтегрального показника

4. На універсальній шкалі вибирається довільна точка  $a = (\bar{X} / \bar{Y})$  і будується її прообраз  $\bar{R}_a = \pi_0^{-1}(a)$  на предметній шкалі  $\bar{R}$ . Далі, відповідно функціям належності  $\mu_{C_i}$  нечітких множин, які відповідають певним термам  $\alpha_i$ , визначаються значення  $\mu_{C_i}(x_a)$ , які приймаються в якості значень відповідних функцій належності  $\mu_{C_i}$  в точці  $a$  на універсальній шкалі. Для всіх значень нечітких змінних, отримаємо функції належності  $\mu_{C_i}$  відповідних значень терм-множин  $\alpha_i$  на універсальній шкалі.

Користуючись моделлю функції належності можна виділити основні правила, за якими формується висновок щодо рівня шумів в цифрових фотоапаратах та представити їх у вигляді нечітких логічних висловлювань:

ЯКЩО  $(X = x_1) \text{ I } (Y = y_1)$ , ТО  $R = \alpha_1$

ЯКЩО  $(X = x_1) \text{ I } (Y = y_2)$  ТО  $R = \alpha_1$

ЯКЩО  $(X = x_1) \text{ I } (Y = y_3)$  ТО  $R = \alpha_2$

ЯКЩО  $(X = x_1) \text{ I } (Y = y_4)$  ТО  $R = \alpha_3$

ЯКЩО  $(X = x_1) \text{ I } (Y = y_5)$  ТО  $R = \alpha_5$

ЯКЩО  $(X = x_2) \text{ I } (Y = y_1)$ , ТО  $R = \alpha_2$

ЯКЩО  $(X = x_2) \text{ I } (Y = y_2)$  ТО  $R = \alpha_3$

ЯКЩО  $(X = x_2) \text{ I } (Y = y_3)$  ТО  $R = \alpha_4$

ЯКЩО  $(X = x_2) \text{ I } (Y = y_4)$  ТО  $R = \alpha_5$

ЯКЩО  $(X = x_2) \text{ I } (Y = y_5)$  ТО  $R = \alpha_5$

## 2.13 Висновок до розділу 2

Другий розділ МКР присвячений математичному моделюванню вимірювання та оцінки шумів, а саме:

- математична модель тестування на шумові завади;
- визначення рівня шумів на основі шумів медіанної фільтрації;
- модель оцінки інтегрального показника;

В результаті проведеного моделювання, а також за матеріалами розділу 1 робимо висновок, що найбільший вплив на величину шуму має якість підсилюючих пристроїв фотокамери, величина зовнішнього освітлення та коефіцієнт підсилення ISO. З врахуванням цього була розроблена нечітка модель логічного виведення частинного інтегрованого показника шуму.

### **3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РЕЙТИНГУВАННЯ ФОТОАПАРАТІВ ЗА ЯКІСТЮ. ОЦІНЮВАННЯ РІВНЯ ШУМУ**

#### **3.1 Розробка алгоритму тестування цифрових фотоапаратів на шумові завади**

Тестування цифрових фотоапаратів на шумові завади проводиться згідно наступного алгоритму.

- 1 Крок. Вибираємо кольорову модель в якій проводиться тестування, наприклад Lab.
- 2 Крок. Задаємо значення шкали чутливості.
- 3 Крок. Вводимо найгірше значення шуму, яке задається користувачем на підставі свого досвіду, або вибираємо автоматичний режим виводу значення.
- 4 Крок. Задаємо значення яскравості.
- 5 Крок. Вибираємо в якому параметрі кольорової моделі Lab провести тестування.
- 6 Крок. Вибираємо файл для тестування.
- 7 Крок. Знаходимо вимірне значення інтенсивності кольору в певному каналі.
- 8 Крок. Знаходимо масив файлу.
- 9 Крок. Визначаємо рівень шуму.
- 10 Крок. Визначаємо оцінку фотоапарата що тестується.
- 11 Крок. Виводиться результат тестування.

Алгоритм, що розглядався, представлений на рисунку 3.1.

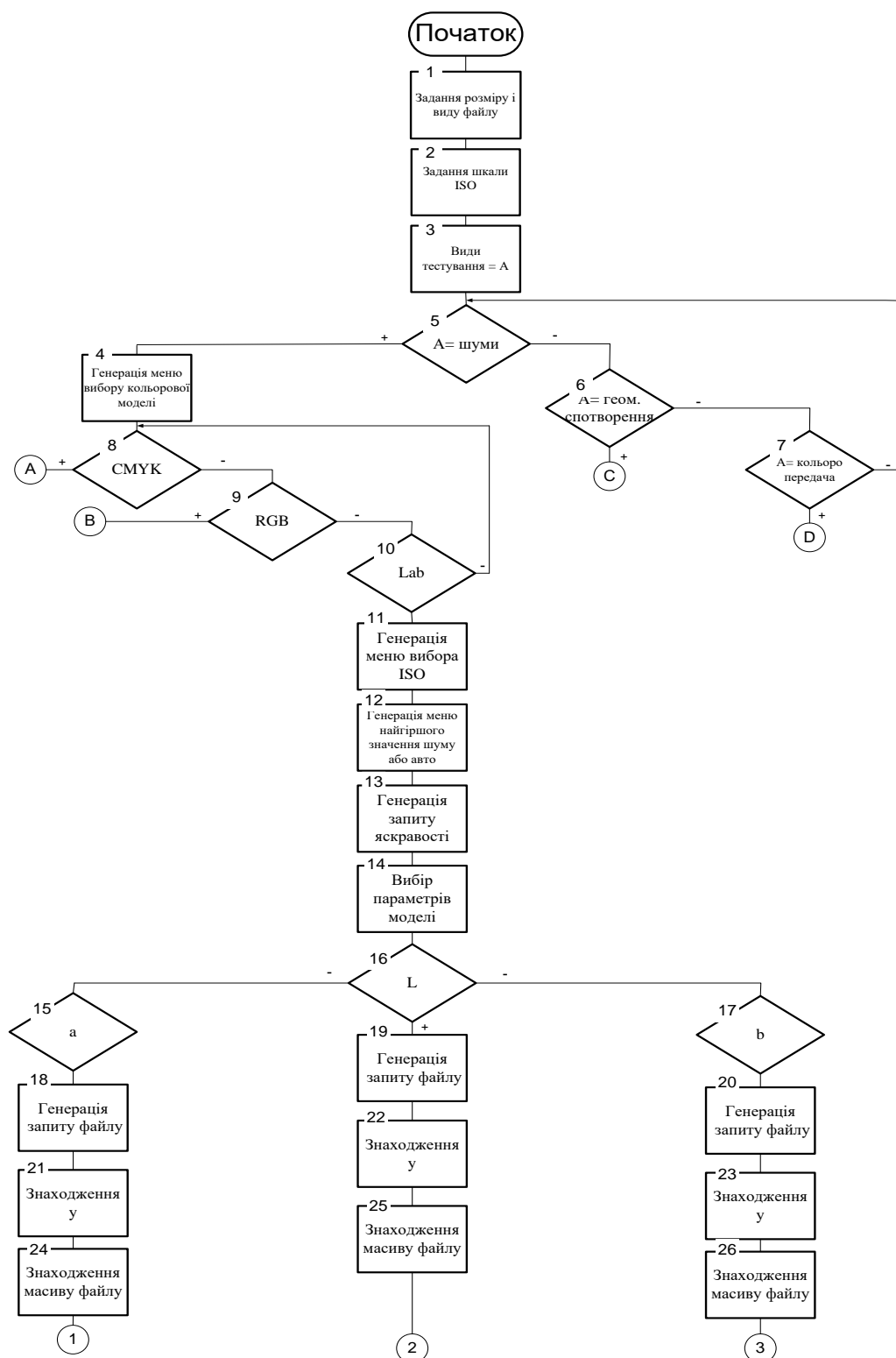
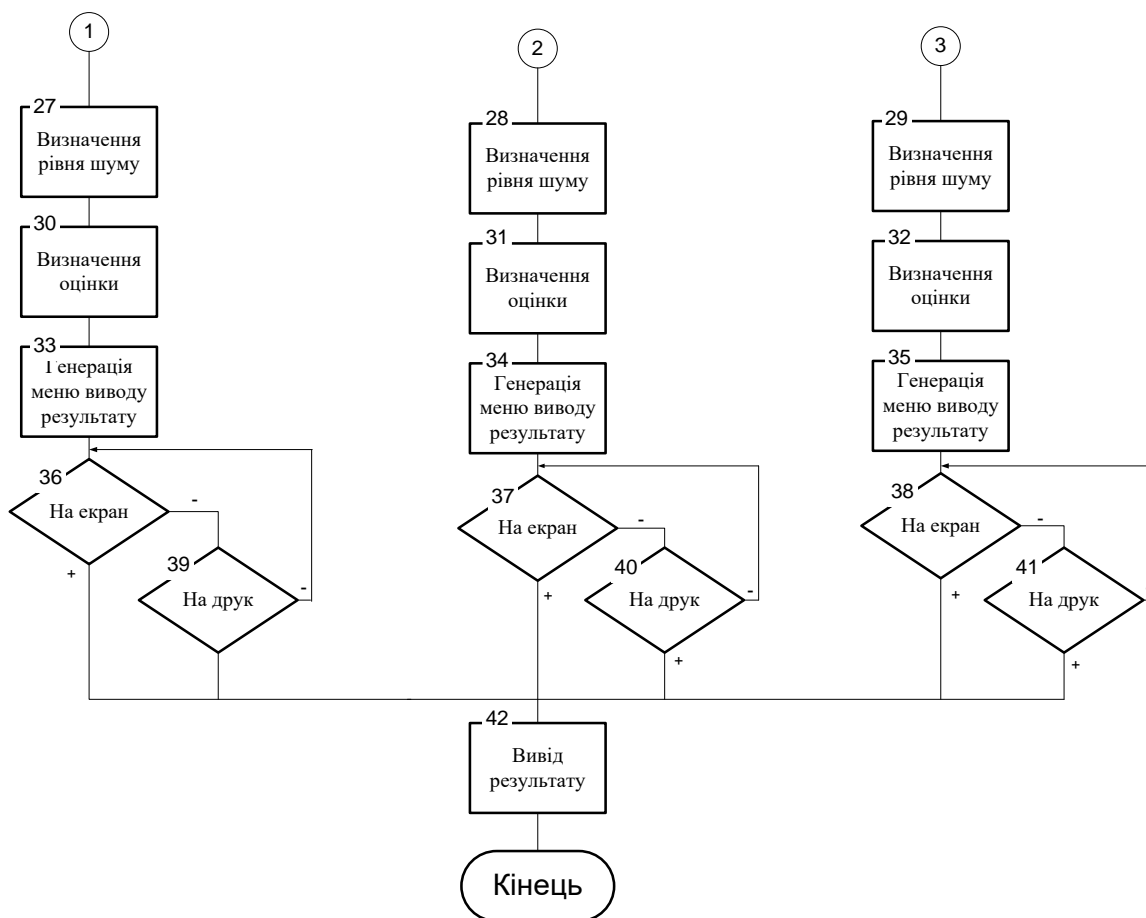


Рисунок 3.1 - Алгоритм тестування на шумові завади

## Продовження рисунку 3.1



### 3.2 Реалізація програмного модуля вимірювання та оцінки рівня шумів

За розробленим у попередньому розділі алгоритмом мовою C++ реалізовано програму, задача якої - зібрати деяку статистику про шуми матриці.

На вхід програми подаються файли, отримані в результаті роботи програми dscaw з параметром-d. На виході отримуємо два текстових файла зі статистикою а також окремі файли із зображеннями шумів у нормалізованому вигляді (тобто за вирахуванням фону плюс зміщені до середнього рівня).

Файли з нормалізованими шумами кладуться в каталог out, який створюється заздалегідь.

Програма обробляє за один раз будь-яку кількість вхідних файлів (тоді результуючі текстові файли будуть містити повні дані і їх залишається тільки імпортувати в яку-небудь електронну таблицю).

Перший вихідний файл – noise\_freq.txt – містить послідовність чисел, що відображають кількість пікселів, у яких зустрічаються певні відхилення від еталонного зображення. Перше число містить кількість пікселів з нульовим відхиленням від еталону, друге число – з відхиленням в одне значення, третє – в два значення і так далі до відхилення в 255. При цьому значення перераховані в діапазон 0 .. 1, де 1.0 це повна кількість пікселів в аналізованій області кадру. Виглядає він так:

```
«0100-0001.pgm» 0,164 0,303 0,231 0,148 0,082 0,040 0,018 0,008 ...
...
«1600-0125.pgm» 0,056 0,110 0,104 0,095 0,085 0,076 0,066 0,058 ...
```

Другий вихідний файл – noise\_median.txt – містить середньостатистичний коефіцієнт шуму по всьому поля кадру (за винятком країв, де обчислення не виробляються). Виглядає він приблизно так:

```
«0100-0001.pgm» 0,77
«0100-0002.pgm» 0,73
«0100-0004.pgm» 0,73
«0100-0008.pgm» 0,69
...
«1600-0015.pgm» 2,90
«1600-0030.pgm» 2,94
«1600-0060.pgm» 2,96
«1600-0125.pgm» 2,79
```

Цей коефіцієнт є середньозважене відхилення пікселя від його «правильного» значення по всьому полю аналізованої області кадру виражене у відсотках до усього діапазону чутливості матриці. Таким чином, за

вищенаведеною таблиці можна сказати, наприклад, що на витримці 1 / 125 і чутливості ISO1600 середньозважене відхилення пікселів від правильного значення становить 2.79%.

В результаті маніпуляцій описаних вище, ми отримуємо набір RAW файлів для різних значень чутливості та витримок. Тепер треба перейменувати файли для того, щоб закодувати параметри експозиції в ім'я файлу, тобто, наприклад, 0100-0001 це чутливість 100, витримка 1с, 1600-0125 це чутливість 1600, витримка 1/125с і так далі.

Тепер необхідно сконвертувати їх в формат PGM, який розуміє програма noiseanal. Для цього беремо dcraw і запускаємо його на всі файли в каталозі:

```
for x in *. Dng; do dcraw-d $ x; done
```

Для Windows це буде:

```
for% x in (*. Dng) do dcraw-d% x
```

Тепер потрібно скомпілювати програму – вона складається всього лише з одного вихідного файлу. Для компілятора gcc командний рядок буде виглядати приблизно так:

```
gcc-s-O3-o noise-anal noise-anal.cpp
```

### **3.3 Розробка алгоритму роботи модуля оцінки якості растрових графічних зображень**

Програмне забезпечення блоку оцінки якості растрових графічних зображень включатиме в себе такі основні частини:

- Інтерфейс користувача, який забезпечує взаємодію користувача з базою знань та логічним блоком. За допомогою інтерфейсу користувача вводяться вхідні ідентифікаційні дані користувача, за якими і проводиться оцінка якості зображення.

- Логічний блок – містить реалізацію алгоритму обчислення усіх вхідних параметрів, а також їх обробку шляхом використання правил нечіткої бази знань;

- Нечітка база знань - містить правила та відповідні висновки щодо даної предметної області, а також повний опис інформаційної моделі. Вона містить такі складові: змінні, тобто вхідні або вихідні параметри системи підтримки прийняття рішень. Вона має ім'я, її максимальне та мінімальне значення. В кожній змінній описаний її власний набір термів, які мають ім'я, та параметри їх функцій належності до даної змінної; правила, кожне з яких містить у своєму складі набір умов та висновок. Для кожної умови задається ім'я вхідної змінної та ім'я терму. Відповідно до такої комбінації змінних і термів робиться висновок для вихідної змінної; функції належності, обчислюють належність введеного користувачем значення до даної множини термів. Для здійснення обчислення вони використовують параметри термів, які вказані в базі знань.

Головною метою розроблюваного програмного модулю є оцінка якості растрових графічних зображень. Програма розроблена мові C#.

Розроблена програма містить такі основні класи, які реалізують процес оцінки якості зображень:

- 1) клас Main – головний клас, що містить опис для графічного інтерфейсу користувача.

- 2) клас Calculate, який містить методи для оцінки якості зображень з використанням нечіткої логіки, а саме:

- fuzzyGeom – метод, який виконує оцінку якості зображення на предмет геометричних спотворень;

- fuzzyKolor – метод, який виконує оцінку якості зображення на предмет кольороспотворень;

- fuzzyNoise – метод, який виконує оцінку якості зображення предмет шумів;

- finalMark – метод, який виконує комплексну оцінку зображення на основі значень, що були отримані за допомогою попередніх методів.



Розроблювана програма повністю виконує покладені на неї функції.

Запропонований алгоритм може бути представлений у вигляді наступної послідовності кроків .

Крок 1. Завантаження головного вікна програми.

Крок 2. Завантаження зображення для тестування.

Крок 3. Тестування картинки на геометричні спотворення.

Крок 4. Тестування картинки на кольороспотворення.

Крок 5. Тестування картинки на наявність шумів.

Крок 6. Визначення оцінки якості зображення.

Крок 7. Виведення результатів тестування.

На рисунку 3.2 зображена схема алгоритму роботи системи оцінювання якості растрових графічних зображень



Рисунок 3.2 – Загальна схема алгоритму роботи програми

На рисунку 3.3 зображена нечітка база знань спрощеного блоку.

№ п/п	ЯКЩО			ТО
	a	b	c	D
1	М	М	М	Висока якість зображення
2	М	С	М	
3	С	М	М	
4	С	С	М	Середня якість зображення
5	М	М	С	
6	С	С	С	
7	В	В	С	Задовільна якість зображення
8	В	М	В	
9	М	В	В	
10	В	С	В	Низька якість зображення
11	С	В	В	
12	В	В	В	

Рисунок 3.3 – Нечітка база знань спрощеного блоку

### 3.4 Висновок до розділу 3

В розділі 3 проведено проектування програмного забезпечення для інформаційної технології оцінювання якості фотокамер:

- розроблений алгоритм роботи блоку вимірювання та оцінки рівня шумів;
- реалізоване програмне забезпечення блоку шумів;
- реалізовано мовою С++ спрощений експериментальний блок виведення оцінки;
- розроблено інструкцію користувача;
- наведено загальну схему алгоритма роботи програми;
- наведено нечітку базу знань спрощеного блоку.

Одержані результати тестування підтверджують адекватність математичних моделей.

## 4 ЕКОНОМІЧНА ЧАСТИНА

### 4.1 Проведення комерційного та технологічного аудиту інформаційної технології рейтингування фотоапаратів за якістю та оцінювання рівня дисторсії та кольороспотворення.

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Для проведення комерційного та технологічного аудиту залучаємо 3-х незалежних експертів, якими є провідні викладачі випускової або спорідненої кафедри.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу здійснюємо із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, а результати зводимо до таблиці 4.1.

Таблиця 4.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу засобу поляриметричного аналізу оптично активних рідни

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами		
Технічна здійсненність концепції	2	2	2
Ринкові переваги (наявність аналогів)	3	1	3
Ринкові переваги (ціна продукту)	2	2	3
Ринкові переваги (технічні властивості)	3	1	2
Ринкові переваги (експлуатаційні витрати)	3	2	3
Ринкові перспективи (розмір ринку)	2	2	2

## Продовження до таблиці 4.1

Ринкові перспективи (конкуренція)	3	2	3
Практична здійсненність (наявність фахівців)	2	2	2
Практична здійсненність (наявність фінансів)	3	2	1
Практична здійсненність (необхідність нових матеріалів)	3	3	3
Практична здійсненність (термін реалізації)	2	2	3
Практична здійсненність (розробка документів)	3	3	2
Сума балів	31	24	29
Середньоарифметична сума балів, СБ	28		

За результатами розрахунків, наведених в таблиці 4.1 робимо висновок про те, що науково-технічний рівень та комерційний потенціал рейтингування фотоапаратів за якістю та оцінювання рівня дисторсії та кольороспотворення – середній.

#### 4.2 Розрахунок витрат на здійснення науково-дослідної роботи

*Витрати на оплату праці.* Належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь-

які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці».

*Основна заробітна плата дослідників.* Витрати на основну заробітну плату дослідників ( $Z_o$ ) розраховують відповідно до посадових окладів працівників, за формулою:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.1)$$

де  $k$  – кількість посад дослідників, залучених до процесу дослідження;  $M_{ni}$  – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;  $T_p$  – число робочих днів в місяці; приблизно  $T_p = (21 \dots 23)$  дні;  $t_i$  – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 4.2.

Таблиця 4.2 – Витрати на заробітну плату дослідників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	20000	909	40	36360
Розробник	15000	682	40	27280
Програміст	15000	682	40	27280
Всього:	90920			

*Додаткова заробітна плата.* Додаткова заробітна плата  $Z_d$  всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як  $(10 \dots 12)\%$  від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = 0,12 \cdot (Z_o + Z_p) = 0,12 \cdot (90920 + 0) = 10910 \text{ грн.}$$

*Відрахування на соціальні заходи.* Нарахування на заробітну плату  $H_{зп}$  розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$H_{зп} = \beta \cdot (Z_o + Z_p + Z_d) =$$

$$= 0,22 \cdot (90920 + 0 + 10910) = 22403 \text{ грн.} \quad (4.2)$$

де  $Z_o$  – основна заробітна плата розробників, грн.;  $Z_p$  – основна заробітна плата робітників, грн.;  $Z_d$  – додаткова заробітна плата всіх розробників та робітників, грн.;  $\beta$  – ставка єдиного внеску на загальнообов’язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

*Розрахунок витрат на матеріали.* Витрати на матеріали  $M$ , що були використані під час виконання даного етапу роботи, розраховуються за формулою:

$$M = \sum_1^n H_i \cdot C_i \cdot K_i, \quad (4.3)$$

де  $H_i$  – кількість матеріалів  $i$ -го виду, шт.;  $C_i$  – ціна матеріалів  $i$ -го виду, грн.;  $K_i$  – коефіцієнт транспортних витрат,  $K_i = (1,1 \dots 1,15)$ ;  $n$  – кількість видів матеріалів.

Зроблені розрахунки зводимо до таблиці 4.3.

Таблиця 4.3 – Матеріали, що використані на розробку

Найменування матеріалів	Ціна за одиницю, грн.	Витрачено	Вартість витрачених комплектуючих, грн.
Папір А4	200	2	400
Всього, з врахуванням коефіцієнта транспортних витрат			440

*Спецустаткування для наукових (експериментальних) робіт.* Вартість спецустаткування визначається за прейскурантом гуртових цін або за даними базових підприємств за відпускними і договірними цінами.

$$V_{\text{спец}} = \sum_1^k C_i \cdot C_{\text{пр.і}} \cdot K_i, \quad (4.4)$$

де  $C_i$  – ціна придбання спецустаткування  $i$ -го виду, грн.;  $C_{\text{пр.і}}$  – кількість одиниць спецустаткування відповідного виду, шт.;  $K_i$  – коефіцієнт транспортних витрат,  $K_i = (1,1 \dots 1,15)$ ;  $n$  – кількість видів спецустаткування. Зроблені розрахунки зводимо до таблиці 4.4.

Таблиця 4.4 – Витрати на придбання спецустаткування

Найменування спецустаткування	Ціна за одиницю, грн.	Витрачено	Вартість спецустаткування, грн.
Стенд для макрозйомки	5000	1	5000
Всього, з врахуванням коефіцієнта транспортних витрат			5500

*Амортизація обладнання.* Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час (чи для) виконання даного етапу роботи.

У спрощеному вигляді амортизаційні відрахування  $A$  в цілому бути розраховані за формулою:

$$A = \frac{C_6}{T_B} \cdot \frac{t}{12}, \quad (4.5)$$

де  $C_6$  – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.;  $t$  – термін використання основного фонду, місяці;  $T_B$  – термін корисного використання основного фонду, роки. Зроблені розрахунки зводимо до таблиці 4.5.

Таблиця 4.5 – Амортизаційні відрахування за видами основних фондів

Найменування	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців	Сума амортизації, грн.
Ноутбук	20000	5	3	1000
Принтер	5000	5	3	25
Всього	1025			

Витрати на електроенергію для науково-виробничих цілей виведені в таблицю 4.6. Витрати на силову електроенергію  $V_e$ , якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

Таблиця 4.6 – Витрати на електроенергію

Найменування обладнання	Потужність, кВт	Тривалість годин роботи
Стенд	0,8	300

$$V_e = \sum \frac{W_i \cdot t_i \cdot C_e \cdot K_{впн}}{ККД} = \frac{0,8 \cdot 300 \cdot 7,5 \cdot 0,91}{0,95} = 1724 \text{ грн.}, \quad (4.6)$$

$W_i$  – встановлена потужність обладнання, кВт;  $t_i$  – тривалість роботи обладнання на етапі дослідження, год.;  $C_e$  – вартість 1 кВт електроенергії, грн.;  $K_{впн}$  – коефіцієнт використання потужності; ККД – коефіцієнт корисної дії обладнання.

*Інші витрати.* До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:



$$I_B = (Z_o + Z_p) \cdot \frac{H_{IB}}{100\%} = (90920 + 0) \cdot \frac{90}{100} = 81828 \text{ грн.}, \quad (4.7)$$

де  $H_{IB}$  – норма нарахування за статтею «Інші витрати».

*Накладні (загальновиробничі) витрати.* До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{HЗВ} = (Z_o + Z_p) \cdot \frac{H_{HЗВ}}{100\%} = (90920 + 0) \cdot \frac{145}{100} = 131834 \text{ грн.}, \quad (4.8)$$

де  $H_{HЗВ}$  – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

*Витрати на проведення науково-дослідної роботи.* Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$\begin{aligned} B_{заг} &= Z_o + Z_p + Z_{дод} + Z_H + M + B_{спец} + A_{обл} + B_e + \\ + I_B + B_{HЗВ} &= 90920 + 0 + 10910 + 22403 + 440 + 5500 + 1025 + \\ &+ 1724 + 81828 + 131834 = 346584 \text{ грн.} \end{aligned} \quad (4.9)$$

*Загальні витрати.* Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{В_{заг}}{\eta} = \frac{346584}{0,7} = 495120 \text{ грн.}, \quad (4.10)$$

де  $\eta$  – коефіцієнт, що характеризує етап виконання науково-дослідної роботи. Оскільки, якщо науково-технічна розробка знаходиться на стадії розробки промислового зразка, то  $\eta=0,7$ .

### **4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором**

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

В даному випадку відбувається розробка засобу, тому основу майбутнього економічного ефекту буде формувати:  $\Delta N$  – збільшення кількості споживачів, яким надається відповідна інформаційна послуга в аналізовані періоди часу;  $N$  – кількість споживачів, яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки;  $Ц_0$  – вартість послуги у році до впровадження інформаційної системи;  $\pm\Delta Ц_0$  – зміна вартості послуги (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу.

Можливе збільшення чистого прибутку у потенційного інвестора  $\Delta\Pi$  для кожного із років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta\Pi = (\pm\Delta Ц_0 \cdot N + Ц_0 \cdot \Delta N_i) \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (4.11)$$

де  $\pm\Delta\Pi$  – зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай, таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки);  $\pm\Delta\Pi_0$  може мати як додатне, так і від’ємне значення (від’ємне – при зниженні ціни відносно року до впровадження цієї розробки, додатне – при зростанні ціни);  $N$  – основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки;  $\Pi_0$  – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році;  $\Pi_6$  – основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження результатів;  $\Delta N$  – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким показником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки);  $\lambda$  – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість становить 20%, а коефіцієнт  $\lambda = 0,8333$ ;  $\rho$  – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати  $\rho = 0,2 \dots 0,5$ ;  $\vartheta$  – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році  $\vartheta = 18\%$ .

Очікуваний термін життєвого циклу розробки 1 рік, тому:

$$\begin{aligned} \Delta\Pi &= ((5700 - 3500) \cdot 1000 - (1000 - 900) \cdot 3500) \cdot 0,8333 \cdot 0,3 \cdot \left(1 - \frac{18}{100}\right) \\ &= 1395160 \text{ грн.} \end{aligned}$$

Далі розраховують приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1+\tau)^t} = \frac{395160}{(1+0,1)^1} = 1359236 \text{ грн.}, \quad (4.12)$$

де  $\Delta\Pi$  – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн.;  $T$  – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки (приймаємо  $T=1$  рік);  $\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,05 \dots 0,15$ ;  $t$  – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Далі розраховують величину початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot 3B = 2 \cdot 495120 = 990240 \text{ грн.} \quad (4.13)$$

де  $k_{\text{інв}}$  – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай  $k_{\text{інв}}=2 \dots 5$ , але може бути і більшим;  $3B$  – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Тоді абсолютний економічний ефект  $E_{\text{абс}}$  або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = ПП - PV = 1359236 - 990240 = 368996 \text{ грн.}, \quad (4.14)$$

де  $ПП$  – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн.;  $PV$  – теперішня вартість початкових інвестицій, грн.

Оскільки  $E_{abc} > 0$ , то можемо припустити про потенційну зацікавленість інвесторів у розробці.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність  $E_B$  або показник внутрішньої норми дохідності вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Внутрішня економічна дохідність інвестицій  $E_B$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою:

$$E_B = \tau_{ж} \sqrt{1 + \frac{E_{abc}}{PV}} = \sqrt[1]{1 + \frac{368996}{990240}} = 1,17, \quad (4.15)$$

де  $T_{ж}$  – життєвий цикл розробки, роки.

Визначимо бар'єрну ставку дисконтування  $\tau_{мін}$ , тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

Мінімальна внутрішня економічна дохідність вкладених інвестицій  $\tau_{мін}$  визначається за формулою:

$$\tau_{мін} = d + f = 0,9 + 0,05 = 0,95, \quad (4.16)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні  $d = 0,9...0,12$ ;  $f$  – показник, що характеризує ризикованість вкладення інвестицій; зазвичай величина  $f = 0,05...0,5$ , але може бути і значно вищою.

Оскільки  $E_B=1,17 > \tau_{\min}=0,95$ , то потенційний інвестор може бути зацікавлений у фінансуванні впровадження науково-технічної розробки та виведенні її на ринок, тобто в її комерціалізації.

Далі розраховуємо період окупності інвестицій  $T_o$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_o = \frac{1}{E_B} = \frac{1}{1,17} = 0,85 \text{ року.} \quad (4.13)$$

Оскільки  $T_o=0,85 < 1 \dots 3$ -х років, то це свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

#### **4.4 Висновок до розділу 4**

Згідно з проведеними дослідженнями рівень комерційного потенціалу розробки за темою «Інформаційна технологія рейтингування фотоапаратів за якістю. Оцінювання рівня шуму фотоапаратів» становить 28 балів, що свідчить про комерційну важливість проведення таких досліджень (рівень комерційного потенціалу середній).

Економічна частина цієї роботи містить розрахунок витрат на розробку інформаційної технології, сума яких складає 495120 гривень. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення та економічний ефект при використанні цієї розробки. Термін окупності становить 0,85 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки.

Отже, можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія рейтингування фотоапаратів за якістю. Оцінювання рівня шуму фотоапаратів».

## ВИСНОВКИ

Всі задачі дослідження виконані в повному об'ємі, а саме:

- проаналізовано існуючі методології вимірювання шумів;
- виконані математичне моделювання та формалізація ієрархічного процесу виведення узагальненого (інтегрованого) показника рівня шумів, інтегральний показник рівня шумів, залежить від освітленості, рівня ISO, величини діафрагми та витримки фотоапарату;
- виконане математичне моделювання вимірювання рівня шумів, рівень шумів при цьому визначається як математичне очікування результатів вимірювання шумів одержаних після медіаної фільтрації;
- спроектований та реалізований алгоритм оцінювання рівня шуму;
- виконані проектування та реалізація програмного модуля технології оцінювання якості фотокамер, його робота базується на нечіткому логічному виводі, за допомогою нечітких логічних рівнянь з нечіткої бази даних;
- було проведено тестування програмного продукту та виконано аналіз отриманих результатів, було зафіксовано коригування функцій належності уже після 3 тестування;
- розроблена інструкція користувача.
- розроблена економічна частина.

Експертні системи, створені за класичними методами, демонструють достовірність оцінювання якості фотокамер на рівні 94-96%. Падіння достовірності з часом складає 3-5% на рік. Запропонована технологія в результаті адаптації демонструє достовірність 96-97% і немає тенденції до падіння достовірності з часом.

Результати роботи відповідають поставленим задачам. Поставлена мета, а саме підвищення достовірності роботи технології оцінювання якості фотокамер, досягнута за рахунок автоматизованого коригування функцій належності нечітких термів по результатам аналізу статистичних даних попередніх вимірювань.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. В. О. Дацюк, Є. О. Михайленко, А. В. Сілагін, Є.О. Сілагін. Інформаційна технологія оцінювання якості фотокамер з використанням нечіткої логіки. Матеріали ІІ Науково-технічна конференції Молодь в науці» (м. Вінниця, Україна, 2023 р)
2. Ротштейн О.П. Інтелектуальні технології ідентифікації: нечіткі множини, генетичні алгоритми, нейронні мережі. – Вінниця: Універсум – Вінниця, 1999. – 320с., іл.
3. Арсенюк І.Р., Кукунін С.В., Сілагін О.В. Застосування апарату нечіткої логіки для оцінки якості графічних растрових зображень // «IES-2014» Зб. наук. Праць. – Вінниця ; ВНТУ, 2014 – 321с.
4. Садиков С.С. Цифрова обробка и аналіз зображень – Ташкент: НПО “Кібернетика” АН Руз – 1994 – 193 с.
5. Русинов М.М. Композиція оптичних систем. Л., «Машинобудування», 1989. – 484с.
6. Волосов Д.С. Фотографічна оптика. М., «Мистецтво», 1971. – 352с.
7. Р. Гойдин. Сучасні цифрові камери. Mode of acess: world wide web. URL: <http://foto&video.ru/index.html> .
8. Садиков С.С. Цифрова обробка и аналіз зображень – Ташкент: НПО “Кібернетика” АН Руз – 1994 – 193 с.
9. Павлідіс Т. Алгоритми машинної графіки и обробки зображень: Перз англ. – М.: Радіо и зв'язок, 1986 – 400 с., іл.
10. С.М. Горський. Кольорова фотографія методом потрійної експозиції. Mode of acess: world wide web. URL: <http://www.loc.gov/exhibits/empire>.
11. Іван Сухарін. Цифрова “дзеркалка” Nikon D100: нове покоління // Фотомагазин.-2002.- №9.-С.24.
12. Андрей Шеклеїн. Тести цифрових фотокамер // Фотомагазин.- 2002.- №10.- С.19.



13. Юрій Сахаревський. Суб'єктивна оцінка цифрових системних камер // Фотомагазин.-2003.- №3.-С.11.
14. Роман Купець. Обробка зображень в цифровій фотографії // Foto & Video.- 2002.-№6.-С.23.
15. Анатолій Алізар. Гістограма яскравості в процесі зйомки і редагування фотографій. // Foto & Video.-2003.-№2.-С.65.
16. Андрій Тушев. Підручник з Adobe Photoshop 5.5. Mode of access: world wide web. URL: <http://www.cl.spb.ru/chtushev/design/index.htm>.
17. Антон Елієнс. Принципи об'єктно-орієнтованої розробки програм. 2-ге видання. – М.: Видавничий дім “Вільямс”, 2002. – 496 с.
18. Навакатікян О.О., Кальніш В.В. Охорона праці користувачів комп'ютерних відеодисплейних терміналів. – К.: Здоров'я, 1997. – 400 с.
19. Правила охорони праці під час експлуатації ЕОМ (Збірник нормативних актів). Навчальний посібник. Уклад. В.П. Якубович. –Вінниця: ВДТУ, 2001. – 79 с.
20. Сергета І.В., Олійник М.П., Бардов В.Г. Гігієна та охорона праці користувача ЕОМ.– -Вінниця: РВВ ВАТ „Віноблдрукарня», 1998. – 112 с.
21. Стрюков С.М. Охорона праці користувачів компютерних відеодисплейних терміналів
22. Навакатікян А.О., Крижановська В.В., Кальніш В.В. Фізіологія і гігієна розумової праці. – К.: Здоров'я, 1987. -; 152 с.
23. Газов В.М.,Соломонов Л.А. Інженерно-психологічне проектування взаємодії людини з технічними засобами. Під ред. В.І.Четверікова. –М.: Вища школа, 1990.
24. Ю. П. Гульчак, Л. І. Северин Основи інженерної психології.- Вінниця ВНТУ, 2004.-105с.

## Додаток А (обов'язковий)

**ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ  
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: Інформаційна технологія рейтингування фотоапаратів за якістю.  
Частина 1. Оцінювання рівня шуму фотоапаратів

Тип роботи: \_\_\_\_\_ магістерська кваліфікаційна робота \_\_\_\_\_  
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІТА  
(кафедра, факультет)

**Показники звіту подібності Unischek**

Оригінальність \_\_\_\_\_ 88,7% \_\_\_\_\_ Схожість \_\_\_\_\_ 11,3% \_\_\_\_\_

**Аналіз звіту подібності (відмітити потрібне):**

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Ознайомлені з повним звітом подібності, який був згенерований системою Unischek щодо роботи.

Автор роботи

Михайленко Є.О.

Керівник роботи

Сілагін О.В.

**Опис прийнятого рішення**

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку

Озеранський В.С.

## Додаток Б (обов'язковий)

### Лістинг програмного коду

TestNoiseFoto.cpp

```

#include "stdafx.h"
#include "DigTest.h"

#include "MainFrm.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CMainFrame

IMPLEMENT_DYNCREATE(CMainFrame, CFrameWnd)

BEGIN_MESSAGE_MAP(CMainFrame, CFrameWnd)
   //{{AFX_MSG_MAP(CMainFrame)
    ON_WM_CREATE()
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()

static UINT indicators[] =
{
    ID_SEPARATOR,
    ID_INDICATOR_CAPS,
    ID_INDICATOR_NUM,
    ID_INDICATOR_SCRL,
};

////////////////////////////////////
// CMainFrame construction/destruction

CMainFrame::CMainFrame()
{
    // TODO: add member initialization code here
}

CMainFrame::~CMainFrame()
{
}

int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)

```

```

{
    if (CFrameWnd::OnCreate(lpCreateStruct) == -1)
        return -1;

    if (!m_wndToolBarStd.CreateEx(this, TBSTYLE_FLAT, WS_CHILD | WS_VISIBLE |
CBRS_TOP
    | CBRS_GRIPPER | CBRS_TOOLTIPS | CBRS_FLYBY | CBRS_SIZE_DYNAMIC) ||
        !m_wndToolBarStd.LoadToolBar(IDR_MAINFRAME))
    {
        TRACE0("Failed to create toolbar\n");
        return -1;    // fail to create
    }

    const int NUM_DRAW_BUTTONS = 3;
    for(int i = 0; i < NUM_DRAW_BUTTONS; i++)
    {
        m_wndToolBarDraw.SetButtonStyle(i,
            m_wndToolBarDraw.GetButtonStyle(i) | TBBS_CHECKGROUP);
    }

    // Set window captions for when toolbars are floating
    m_wndToolBarStd.SetWindowText("Standard");
    m_wndToolBarDraw.SetWindowText("Drawing");

    if (!m_wndStatusBar.Create(this) ||
        !m_wndStatusBar.SetIndicators(indicators,
            sizeof(indicators)/sizeof(UINT)))
    {
        TRACE0("Failed to create status bar\n");
        return -1;    // fail to create
    }

BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)
{
    if( !CFrameWnd::PreCreateWindow(cs) )
        return FALSE;
    return TRUE;
}

////////////////////////////////////
// CMainFrame diagnostics

#ifdef _DEBUG
void CMainFrame::AssertValid() const
{
    CFrameWnd::AssertValid();
}

void CMainFrame::Dump(CDumpContext& dc) const

```

```

{
    CFrameWnd::Dump(dc);
}

#endif //_DEBUG

/////////////////////////////////////////////////////////////////
// CMainFrame message handlers

BOOL CMainFrame::OnCreateClient(LPCREATESTRUCT lpcs,
                                CCreateContext* pContext)
{
    // TODO: Add your specialized code here and/or call the base class

    return m_wndSplitter.Create(this, 2, 2, CSize(1,1),
        pContext);
}

```

Зміст файлу Dialogs.cpp

```

#include "stdafx.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////
// CTitleDialog dialog

CTitleDialog::CTitleDialog(CWnd* pParent /*=NULL*/)
    : CDialog(CTitleDialog::IDD, pParent)
{
   //{{AFX_DATA_INIT(CTitleDialog)
    m_strDrawing = _T("");
    //}}AFX_DATA_INIT
}

void CMyDrawView::OnToolSettings()
{
    // TODO: Add your command handler code here
    // Create property sheet and the 3 contained pages.
    CPropertySheet settings("MyDraw Settings");
    CLinePage dlgLine;
    CShapePage dlgShape;

    // Add the pages to the sheet, in desired order.
    settings.AddPage(&dlgLine);
    settings.AddPage(&dlgShape);
}

```



```
IMPLEMENT_DYNCREATE(CLinePage, CPropertyPage)
```

```
CLinePage::CLinePage() : CPropertyPage(CLinePage::IDD)
{
   //{{AFX_DATA_INIT(CLinePage)
    m_nThickness = 0;
   //}}AFX_DATA_INIT
}
```

```
CLinePage::~CLinePage()
{
}
```

```
void CLinePage::DoDataExchange(CDataExchange* pDX)
{
    CPropertyPage::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CLinePage)
    DDX_Control(pDX, IDC_SPIN1, m_spinThickness);
    DDX_Control(pDX, IDC_COMBO1, m_comboColor);
    DDX_Text(pDX, IDC_EDIT1, m_nThickness);
    DDV_MinMaxUInt(pDX, m_nThickness, 0, 5);
   //}}AFX_DATA_MAP
}
```

```
BEGIN_MESSAGE_MAP(CLinePage, CPropertyPage)
   //{{AFX_MSG_MAP(CLinePage)
    ON_CBN_SELCHANGE(IDC_COMBO1, OnSelchangeComboColor)
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()
```

```
////////////////////////////////////
// CLinePage message handlers
////////////////////////////////////
// CShapePage property page
```

```
IMPLEMENT_DYNCREATE(CShapePage, CPropertyPage)
```

```
CShapePage::CShapePage() : CPropertyPage(CShapePage::IDD)
{
   //{{AFX_DATA_INIT(CShapePage)
    m_bTransparent = FALSE;
    m_nShpRectangle = -1;
   //}}AFX_DATA_INIT
}
```

```
CShapePage::~CShapePage()
{
}
```

```
void CShapePage::DoDataExchange(CDataExchange* pDX)
```

```

{
    CPropertyPage::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CShapePage)
    DDX_Check(pDX, IDC_CHECK1, m_bTransparent);
    DDX_Radio(pDX, IDC_RADIO1, m_nShpRectangle);
   //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CShapePage, CPropertyPage)
   //{{AFX_MSG_MAP(CShapePage)
    // NOTE: the ClassWizard will add message map macros here
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CShapePage message handlers

BOOL CLinePage::OnSetActive()
{
    m_spinThickness.SetRange(0, 5);
    for(int i = 0; i < 11; i++)
    {
        m_comboColor.AddString(arColorNames[i]);
    }
    m_comboColor.SetCurSel(m_nColorIndex);

    return CPropertyPage::OnSetActive();
}

void CLinePage::OnSelchangeComboColor()
{
    m_nColorIndex = m_comboColor.GetCurSel();
    m_colorRGB = arColors[m_nColorIndex];
    if(m_nColorIndex == CUSTOM_COLOR_INDEX)
    {
        GetCustomColor();
    }
}

void CLinePage::GetCustomColor()
{
    CColorDialog dlg;
    dlg.m_cc.rgbResult = m_colorRGB;
    if(dlg.DoModal() == IDOK)
    {
        m_colorRGB = dlg.GetColor();
        m_nColorIndex = CUSTOM_COLOR_INDEX;
    }
}

```



```
}
```

Зміст файлу Init.cpp

```
#include "stdafx.h"
#include "MainFrm.h"

#include "DigTest.h"
#include "DrawVw.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

BEGIN_MESSAGE_MAP(CDigTestApp, CWinApp)
//{{AFX_MSG_MAP(CDigTestApp)
ON_COMMAND(ID_APP_ABOUT, OnAppAbout)
// NOTE - the ClassWizard will add and remove mapping macros here.
// DO NOT EDIT what you see in these blocks of generated code!
//}}AFX_MSG_MAP
// Standard file based document commands
ON_COMMAND(ID_FILE_NEW, CWinApp::OnFileNew)
ON_COMMAND(ID_FILE_OPEN, CWinApp::OnFileOpen)
// Standard print setup command
ON_COMMAND(ID_FILE_PRINT_SETUP, CWinApp::OnFilePrintSetup)
END_MESSAGE_MAP()

CDigTestApp::CDigTestApp()
{
}

CDigTestApp theApp;

////////////////////////////////////
// CDigTestApp initialization

BOOL CDigTestApp::InitInstance()
{

#ifdef _AFXDLL
    Enable3dControls();
#else
    Enable3dControlsStatic();
#endif
}
#endif
```

```

SetRegistryKey(_T("Local AppWizard-Generated Applications"));

LoadStdProfileSettings();

CSingleDocTemplate* pDocTemplate;
pDocTemplate = new CSingleDocTemplate(
    IDR_MAINFRAME,
    RUNTIME_CLASS(CDigTestDoc),
    RUNTIME_CLASS(CMainFrame),
    RUNTIME_CLASS(CDigTestView));
AddDocTemplate(pDocTemplate);

    Enable DDE Execute open
EnableShellOpen();
RegisterShellFileTypes(TRUE);

CCommandLineInfo cmdInfo;
ParseCommandLine(cmdInfo);

    Dispatch commands specified on the command line
if (!ProcessShellCommand(cmdInfo))
    return FALSE;

    The one and only window has been initialized, so show and update it.
m_pMainWnd->ShowWindow(SW_SHOW);
m_pMainWnd->UpdateWindow();

m_pMainWnd->DragAcceptFiles();

return TRUE;
}

class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

   //{{AFX_DATA(CAboutDlg)
enum { IDD = IDD_ABOUTBOX };
//}}AFX_DATA

   //{{AFX_VIRTUAL(CAboutDlg)
protected:
virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
//}}AFX_VIRTUAL

protected:
   //{{AFX_MSG(CAboutDlg)
// No message handlers
//}}AFX_MSG
DECLARE_MESSAGE_MAP()

```

```

};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
   //{{AFX_DATA_INIT(CAboutDlg)
   //}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CAboutDlg)
   //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
   //{{AFX_MSG_MAP(CAboutDlg)
    // No message handlers
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()

void CDigTestApp::OnAppAbout()
{
    CAboutDlg aboutDlg;
    aboutDlg.DoModal();
}

```

Зміст файлу Shape.cpp: імплементація класів для обробки зображень.

```

#include "stdafx.h"
#include "DigTest.h"
#include "Shape.h"
#include "Resource.h"

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

////////////////////////////////////
// Construction/Destruction
////////////////////////////////////

IMPLEMENT_SERIAL(CShape, CObject, 1)

CTestNoise:: CTestNoise ()
{
    m_boxShape.SetRect(0, 0, 0, 0);
    m_bTransparent = true;
}

```

```

    m_nColorShape = ID_COLOR_BLACK;
    m_bSelected = false;
}

CTestNoise::~CTestNoise ()
{
}

IMPLEMENT_SERIAL(CTestResolution, CShape, 1)

Void CTestResolution::Draw(CDC* pDC, bool bSelectionModeOn) // Virtual override
{
    pDC->Rectangle(m_boxShape);
    if(m_bSelected && bSelectionModeOn)
        DrawHandles(pDC);
}

IMPLEMENT_SERIAL(CTestDiffusion, CShape, 1)

void CTestDiffusion::Draw(CDC* pDC, bool bSelectionModeOn) {
    pDC->Ellipse(m_boxShape);
    if(m_bSelected && bSelectionModeOn)
        DrawHandles(pDC);
}

void CShape::Serialize(CArchive& ar)
{
    BYTE byTransparent;

    CObject::Serialize(ar);

    if (ar.IsStoring())
    {
        byTransparent = (m_bTransparent ? 1 : 0);
        ar << m_boxShape << byTransparent << m_nColorShape;
    }
    else
    {
        ar >> m_boxShape >> byTransparent >> m_nColorShape;
        m_bTransparent = (byTransparent != 0);
        m_bSelected = false;    }
}

void CShape::CreateHandleRects()
{
    // Calculate the rectangles for a set of selection handles.
    // Store them in array of handle rects.

```

```

int nHandSize = GetSystemMetrics(SM_CXBORDER) * 7;

// Create an inflated rect around the shape's bounding
// rect, m_boxShape.
CRect ri = m_boxShape;
ri.InflateRect(nHandSize, -nHandSize);

CRect rectLeftTop(ri.left, ri.top,
                 m_boxShape.left, m_boxShape.top);
rectLeftTop.OffsetRect(4, -4);
arHandles[0] = rectLeftTop;

CRect rectRightTop(m_boxShape.right, ri.top,
                  ri.right, m_boxShape.top);
rectRightTop.OffsetRect(-4, -4);
arHandles[1] = rectRightTop;

CRect rectRightBottom(m_boxShape.right, m_boxShape.bottom,
                     ri.right, ri.bottom);
rectRightBottom.OffsetRect(-4, 4);
arHandles[2] = rectRightBottom;

CRect rectLeftBottom(ri.left, m_boxShape.bottom,
                    m_boxShape.left, ri.bottom);
rectLeftBottom.OffsetRect(4, 4);
arHandles[3] = rectLeftBottom;
int centerVert = ri.left + (ri.right - ri.left) / 2;
int leftVert = centerVert - (nHandSize / 2);
int rightVert = centerVert + (nHandSize / 2);

CRect rectBottomCenter(leftVert, ri.top, rightVert,
                       m_boxShape.top);
rectBottomCenter.OffsetRect(0, -4);
arHandles[4] = rectBottomCenter;

CRect rectTopCenter(leftVert, m_boxShape.bottom,
                   rightVert, ri.bottom);
rectTopCenter.OffsetRect(0, 4);
arHandles[6] = rectTopCenter;
int centerHorz = ri.top + (ri.bottom - ri.top) / 2;
int topHorz = centerHorz - (nHandSize / 2);
int bottomHorz = centerHorz + (nHandSize / 2);

CRect rectRightCenter(m_boxShape.right, topHorz,
                     ri.right, bottomHorz);
rectRightCenter.OffsetRect(-4, 0);
arHandles[5] = rectRightCenter;
CRect rectLeftCenter(ri.left, topHorz, m_boxShape.left,
                    bottomHorz);
rectLeftCenter.OffsetRect(4, 0);
arHandles[7] = rectLeftCenter;

```

```

}

void CTestColor::ColorHandles(CDC *pDC)
{
    CreateHandleRects();
    for(int i = 0; i < 8; i++)
    {
        pDC->Rectangle(arHandles[i]);
    }
    pDC->SelectObject(pBrushOld);
}
}

```

Зміст файлу Result.h

```

#if
!defined(AFX_DRAWDOC_H__24958324_5D0A_11D2_991B_00C04FC29F5C__INCLUDED_)
#define AFX_DRAWDOC_H__24958324_5D0A_11D2_991B_00C04FC29F5C__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#include "Shape.h"

class CMyDrawDoc : public CDocument
{
protected: // create from serialization only
    CMyDrawDoc();
    DECLARE_DYNCREATE(CMyDrawDoc)

// Attributes
public:
    void SetToOldestShape() { m_pos = m_listShapes.GetTailPosition(); };
    void SetToLatestShape() { m_pos = m_listShapes.GetHeadPosition(); };

    CShape* GetPrevShape()
    {
        // Sets m_pos to NULL if no shapes or if
        // latest shape is last in list.
        return (CShape*)m_listShapes.GetPrev(m_pos);
    };

    CShape* GetNextShape()
    {
        // Sets m_pos to NULL if no shapes or if
        // latest shape is last in list.
        return (CShape*)m_listShapes.GetNext(m_pos);
    };
}

```

```

POSITION GetPos() const
{
    // m_pos tells you where you are in a list of the shapes.
    // Use GetPos with either iteration direction to test for end.
    return m_pos; // Can be NULL
};

int GetCount() const
{
    // Return the number of stored shapes.
    return m_listShapes.GetCount();
};

private:
    COBList m_listShapes; // Linked list of all shapes drawn so far
    POSITION m_pos; // Latest position accessed

// Operations
public:

// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CMyDrawDoc)
public:
    virtual BOOL OnNewDocument();
    virtual void Serialize(CArchive& ar);
    virtual void DeleteContents();
//}}AFX_VIRTUAL

// Implementation
public:
    void GetDocSizes(int nMapMode, CSize& sizeTotal, CSize& sizePage, CSize& sizeLine);
    CShape* CreateShape(ShpType st);
    void DeleteAllShapes();
    void DeleteLatestShape();
    virtual ~CMyDrawDoc();
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected:

// Generated message map functions
protected:
   //{{AFX_MSG(CMyDrawDoc)
    // NOTE - the ClassWizard will add and remove member functions here.
    // DO NOT EDIT what you see in these blocks of generated code !
//}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

```

```

////////////////////////////////////

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif //
!defined(AFX_DRAWDOC_H__24958324_5D0A_11D2_991B_00C04FC29F5C__INCLUDED_)

Зміст файлу RGB.h

// DrawVw.h : interface of the CMyDrawView class
//
////////////////////////////////////

#if
!defined(AFX_DRAWVW_H__24958326_5D0A_11D2_991B_00C04FC29F5C__INCLUDED_)
#define AFX_DRAWVW_H__24958326_5D0A_11D2_991B_00C04FC29F5C__INCLUDED_

#if _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

#include "DrawDoc.h"
#include "dialogs.h"

// Array of actual colors, indexed
// by CMyDrawView::m_nColorNext
static COLORREF arColors[11] =
{
    RGB(0,0,0),    // Black
    RGB(0,0,255), // Blue
    RGB(0,255,0), // Green
    RGB(0,255,255), // Cyan
    RGB(255,0,0), // Red
    RGB(255,0,255), // Magenta
    RGB(255,255,0), // Yellow
    RGB(255,255,255), // White
    RGB(128,128,128), // Dark gray
    RGB(192,192,192), // Light gray
    RGB(0,0,0)    // Custom color
};

const UINT CUSTOM_COLOR_INDEX = 10;
const UINT CUSTOM_COLOR = ID_COLOR_BLACK + CUSTOM_COLOR_INDEX;

class CMyDrawView : public CScrollView
{
protected: // create from serialization only
    CMyDrawView();
    DECLARE_DYNCREATE(CMyDrawView)
}

```



```

// Attributes
public:
    CMyDrawDoc* GetDocument();
    ShpType m_typeNext;
    CTestNoise* m_pShpTemp;
    bool m_bCaptured;
    CRes* m_pResOld;
    bool m_bTransparent;
    UINT m_nColorNext;
    int m_nMapMode;
    int m_nPageWidth;
    int m_nPageHeight;
    CString m_strDrawing;
    bool m_bSelectionMode;

// Operations
public:

// Overrides
// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CMyDrawView)
public:
    virtual void OnDraw(CDC* pDC);
    virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
    virtual void OnInitialUpdate();
    virtual void OnPrepareDC(CDC* pDC, CPrintInfo* pInfo = NULL);
protected:
    virtual BOOL OnPreparePrinting(CPrintInfo* pInfo);
    virtual void OnBeginPrinting(CDC* pDC, CPrintInfo* pInfo);
    virtual void OnEndPrinting(CDC* pDC, CPrintInfo* pInfo);
    virtual void OnPrint(CDC* pDC, CPrintInfo* pInfo);
//}}AFX_VIRTUAL

// Implementation
public:
    void DoHitTesting(CPoint point);
    void CancelSelection();
    void PrintFooter(CDC *pDC, CPrintInfo *pInfo);
    void PrintHeader(CDC* pDC, CPrintInfo* pInfo);
    void Invert (CDC *pDC, CShape &s, bool bInvert = true);

#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif

protected:

// Generated message map functions
protected:
   //{{AFX_MSG(CMyDrawView)

```

```

afx_msg void OnToolRectangle();
afx_msg void OnToolEllipse();
afx_msg void OnLButtonDown(UINT nFlags, CPoint point);
afx_msg void OnMouseMove(UINT nFlags, CPoint point);
afx_msg void OnLButtonUp(UINT nFlags, CPoint point);
afx_msg void OnUpdateToolRectangle(CCmdUI* pCmdUI);
afx_msg void OnUpdateToolEllipse(CCmdUI* pCmdUI);
afx_msg void OnToolTransparent();
afx_msg void OnUpdateToolTransparent(CCmdUI* pCmdUI);
afx_msg void OnToolSelection();
afx_msg void OnUpdateToolSelection(CCmdUI* pCmdUI);
afx_msg void OnEditDrawingTitle();
afx_msg void OnToolCustomColor();
afx_msg void OnUpdateToolCustomColor(CCmdUI* pCmdUI);
afx_msg void OnToolSettings();
//}AFX_MSG
afx_msg void OnToolColor(UINT nID); // ON_COMMAND_RANGE handler
afx_msg void OnUpdateToolColor(CCmdUI* pCmdUI);
DECLARE_MESSAGE_MAP()
};

#ifdef _DEBUG // debug version in DrawVw.cpp
inline CMyDrawDoc* CMyDrawView::GetDocument()
{ return (CMyDrawDoc*)m_pDocument; }
#endif

////////////////////////////////////

//{{AFX_INSERT_LOCATION}}
// Microsoft Visual C++ will insert additional declarations immediately before the previous line.

#endif //
!defined(AFX_DRAWVW_H__24958326_5D0A_11D2_991B_00C04FC29F5C__INCLUDED_)

Зміст файлу Colors.cpp

#ifdef AFX_SHAPE_H__97288485_7254_11D2_991B_00C04FC29F5C__INCLUDED_
#define AFX_SHAPE_H__97288485_7254_11D2_991B_00C04FC29F5C__INCLUDED_

#ifdef _MSC_VER > 1000
#pragma once
#endif // _MSC_VER > 1000

enum ShpType
{
    shpSelecting,
    shpRectangle,
};

static CRect arHandles[8];

static CString arColorNames[11] =

```

```

{
    "Black",
    "Blue",
    "Green",
    "Cyan",
    "Red",
    "Magenta",
    "Yellow",
    "White",
    "Dark Gray",
    "Light Gray",
    "Custom Color"
};

class CTest : public CObject
{
public:
    DECLARE_SERIAL(CShape)

    CTest();
    CTest (const CTest& s)
    {
        m_boxShape = s.m_boxShape;
        m_bTransparent = s.m_bTransparent;
        m_nColorShape = s.m_nColorShape;
        m_bSelected = s.m_bSelected;
    }

    CTest& operator=(const CTest& s)
    {
        return *this;
    }

    virtual void Result(CDC* pDC, bool bSelectionModeOn)
        { TRACE("My Error: In Test::Test.\n");
          ASSERT(FALSE); };
    virtual void Serialize(CArchive& ar);

// Implementation
public:
    void DrawHandles(CDC *pDC);
    void CreateHandleRects();
    virtual ~CShape();

};

// Concrete subclass of abstract base class CShape
class CShpRectangle : public CShape
{

```

```

public:
    DECLARE_SERIAL(CTestDiff)

    void Diffusion(CDC* pDC, bool bSelectionModeOn);

// Implementation
public:
};

    void Adapter(CDC* pDC, bool bSelectionModeOn);

// Implementation
public:
    void CreateHandleRects();
};

#endif // !defined(AFX_SHAPE_H__97288485_7254_11D2_991B_00C04FC29F5C__INCL

```

Зміст файлу Colors.cpp

```

#include <graph.h>
#include <math.h>
#include <string.h>

#define PI    3.141593

long cvlColors[256];

struct TURTLE tc = { 1.39 };

short InitTurtle( struct videoconfig *vc )
{
    int i;
    unsigned cvuInc, cvuRed, cvuGreen, cvuBlue;
    static int mode = -1;

    if( !vc->numxpixels )
        return 0;

    if( mode != vc->mode )
    {
        mode = vc->mode;
        tc.xsLeft = tc.ysTop = 0;
        tc.xsRight = vc->numxpixels - 1;
        tc.ysBot = vc->numypixels - 1;
    }
}

```

```

switch( vc->adapter )
{
    case _MDPA:
    case _CGA:
    case _OCGA:
    case _HGC:
        tc.fPalette = FALSE;
        break;
    default:
        tc.fPalette = TRUE;
        break;
}

switch( vc->mode )
{
    case _HRESBW:
    case _HERCMONO:
    case _ERESNOCOLOR:
    case _ORESCOLOR:
    case _VRES2COLOR:
        tc.ccv = 0;
        tc.cci = 2;
        return Home( vc );
    case _MRES256COLOR:
        cvuInc = 12;
        tc.ccv = tc.cci = 125;
        break;
    case _ERESCOLOR:
        if( vc->memory == 64 )
        {
            cvuInc = 32;
            tc.ccv = 16;
            tc.cci = 4;
            break;
        }
    case _VRES16COLOR:
        cvuInc = 16;
        tc.ccv = 64;
        tc.cci = 16;
        break;
    case _MRES4COLOR:
    case _MRESNOCOLOR:
        cvuInc = 32;
        tc.ccv = 16;
        tc.cci = 4;
        break;
    case _MRES16COLOR:
    case _HRES16COLOR:
        cvuInc = 32;
        tc.cci = tc.ccv = 16;
        break;
}

```

```

for( i = 0, cvuBlue = 0; cvuBlue < 64; cvuBlue += cvuInc )
  for( cvuGreen = 0; cvuGreen < 64; cvuGreen += cvuInc )
    for( cvuRed = 0; cvuRed < 64; cvuRed += cvuInc )
      {
        cvlColors[i] = RGB( cvuRed, cvuGreen, cvuBlue );

        if( cvuInc == 32 )
          cvlColors[i + 8] = cvlColors[i] | (cvlColors[i] >> 1);
          i++;
      }
    cvlColors[tc.ccv - 1] = _BRIGHTWHITE;
    NextColorValue( DEFAULT );
    return Home( vc );
}

short Home( struct videoconfig *vc )
{
  struct _wxycoord xy1, xy2;

  _setviewport( tc.xsLeft, tc.ysTop, tc.xsRight, tc.ysBot );

  tc.yMax = 500.0;
  tc.xMax = tc.yMax * tc.yxRatio;
  if( !_setwindow( FALSE, -tc.xMax, -tc.yMax, tc.xMax, tc.yMax ) )
    return 0;

  xy1 = _getwindowcoord( 1, 1 );
  xy2 = _getwindowcoord( 1, 2 );
  tc.yUnit = xy2.wy - xy1.wy;

  tc.xCur = tc.yCur = 0.0;
  _moveto_w( tc.xCur, tc.yCur );
  TurnTo( 0 );
  PenDown( TRUE );
  SetFill( FALSE );

  _remappalette( WHITE, _BRIGHTWHITE );
  BorderColor( WHITE );
  PenColor( WHITE );
  _setbkcolor( _BLACK );
  return 1;
}

int PenDown( int fPenDown )
{
  switch( fPenDown )

```

```

{
  case DEFAULT:
    break;
  case FALSE:
    tc.fPenDown = FALSE;
    break;
  default:
    tc.fPenDown = TRUE;
    break;
}
return tc.fPenDown;
}

```

```

short SetFill( short fFill )
{
  switch( fFill )
  {
    case DEFAULT:
      break;
    case _GBORDER:
    case FALSE:
      tc.fFill = _GBORDER;
      break;
    default:
      tc.fFill = _GFILLINTERIOR;
      break;
  }
  return tc.fFill;
}

```

```

short BorderColor( short border )
{
  if( border != DEFAULT )
    tc.ciBorder = border;
  return tc.ciBorder;
}

```

```

short Turn( short angCur )
{
  return( tc.angCur = ((tc.angCur + angCur) % CIRCUMFERENCE) );
}

```

```

short TurnTo( short angCur )
{
  if( angCur < 0 )
    return( tc.angCur = 360 - (angCur % CIRCUMFERENCE) );
  else
    return( tc.angCur = angCur % CIRCUMFERENCE );
}


```

## Додаток В (обов'язковий)

## ІЛЮСТРАТИВНА ЧАСТИНА

«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РЕЙТИНГУВАННЯ ФОТОАПАРАТІВ ЗА ЯКІСТЮ. ЧАСТИНА 1. ОЦІНЮВАННЯ РІВНЯ ШУМУ ФОТОАПАРАТІВ»

Виконав: студент 2-го курсу, групи 1КН-22м  
спеціальності 122 – Комп'ютерні науки



Михайленко Є.О.

(прізвище та ініціали)

Керівник: к.т.н., доцент кафедри КН



Сілагін О. В.

(прізвище та ініціали)

« 07 » 12 2023 р.



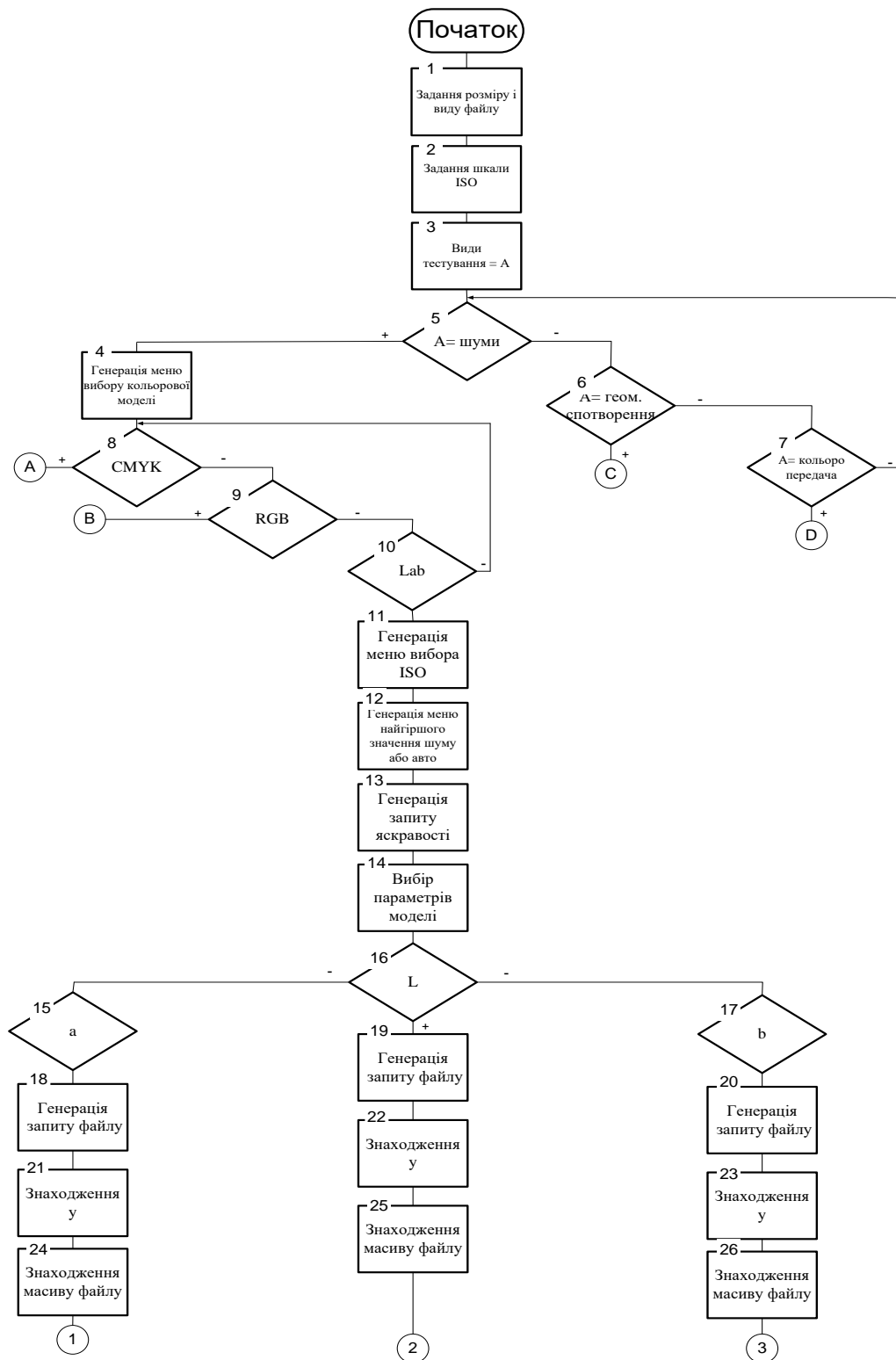


Рисунок В.1 – алгоритм тестування на шумові завади



Рисунок В.2 – загальна схема алгоритму роботи програми

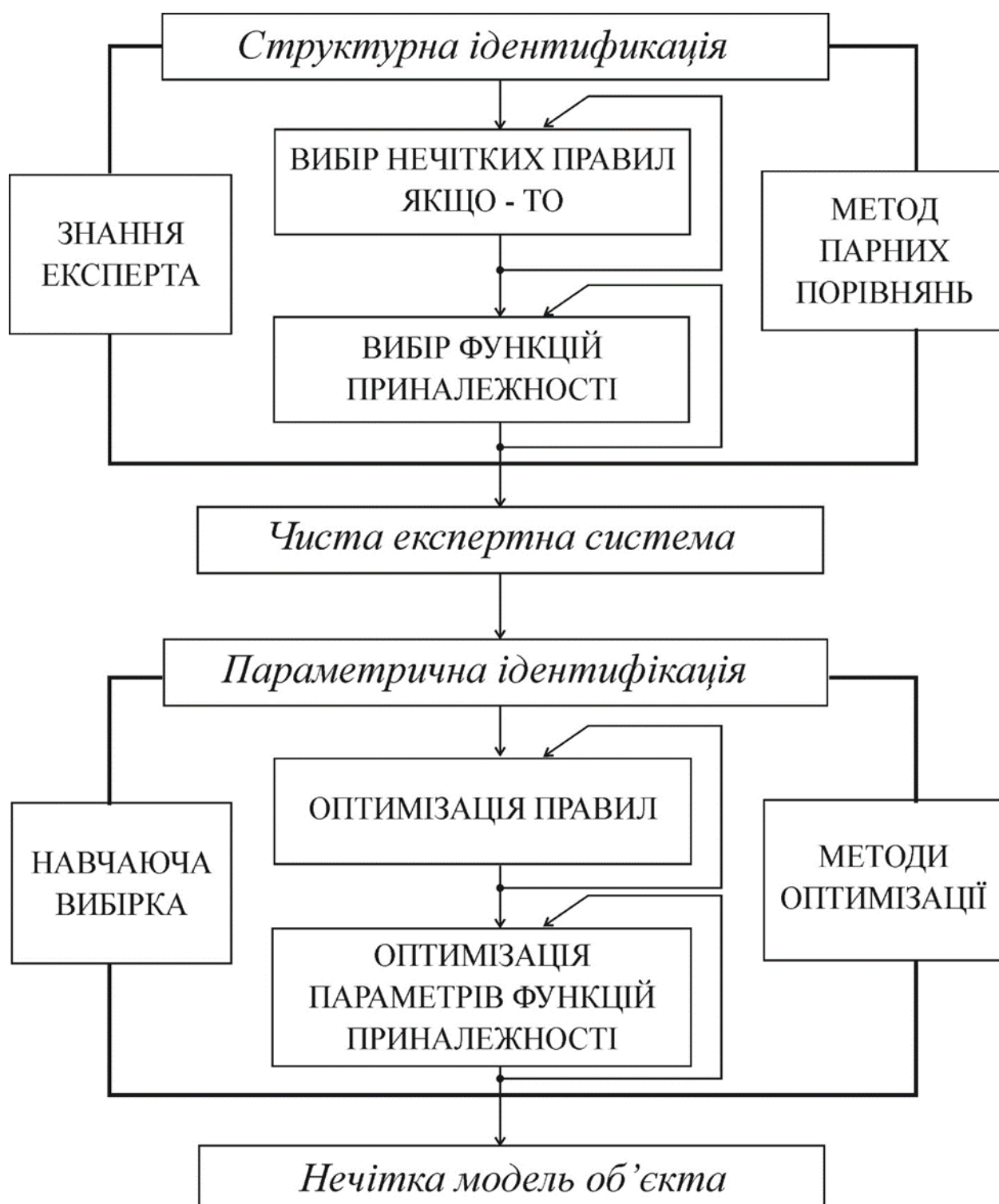


Рисунок В.3 – Двоетапна модель налаштування нечіткої системи

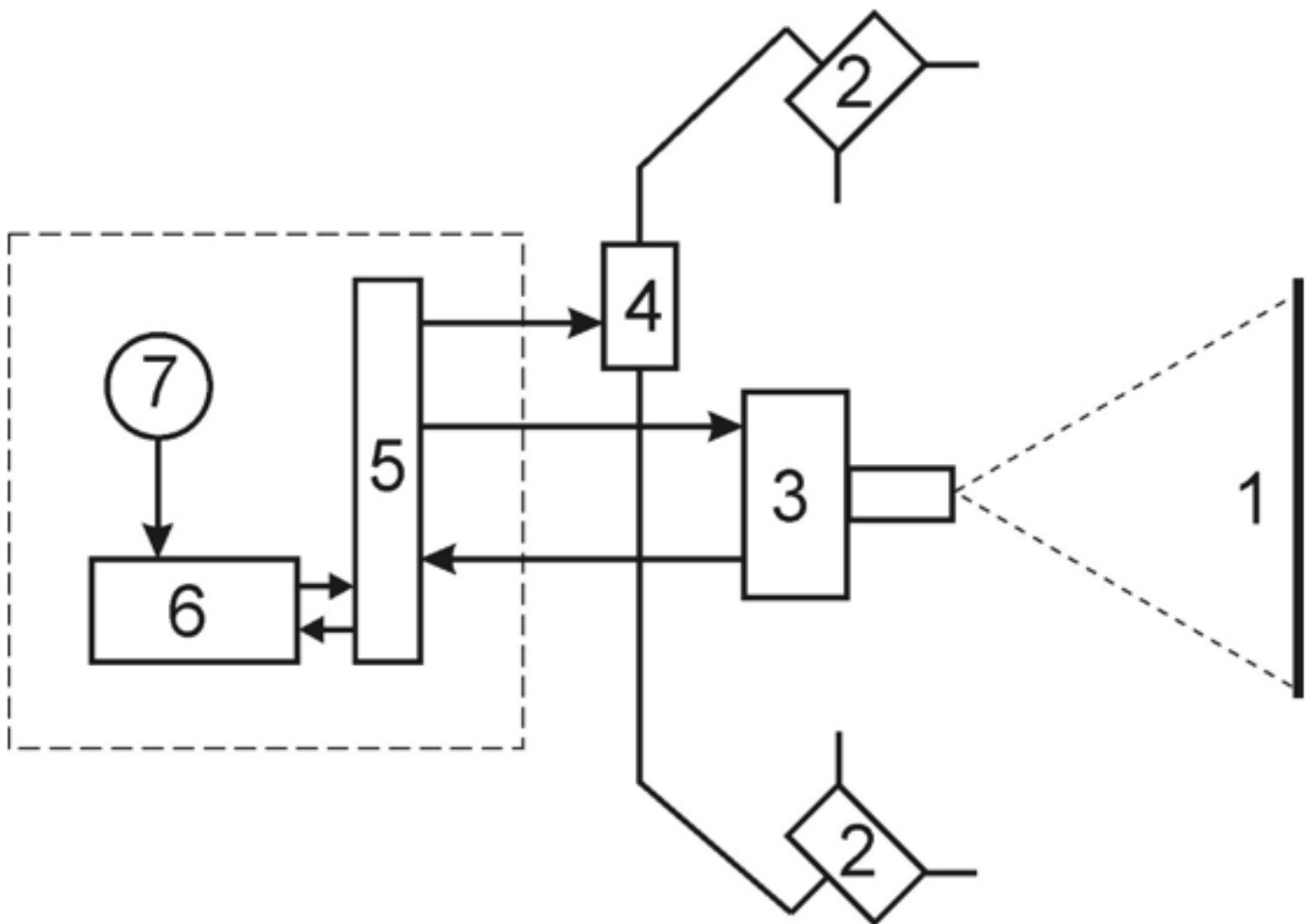


Рисунок В.4 – Структурна схема системи оцінки якості растрових зображень

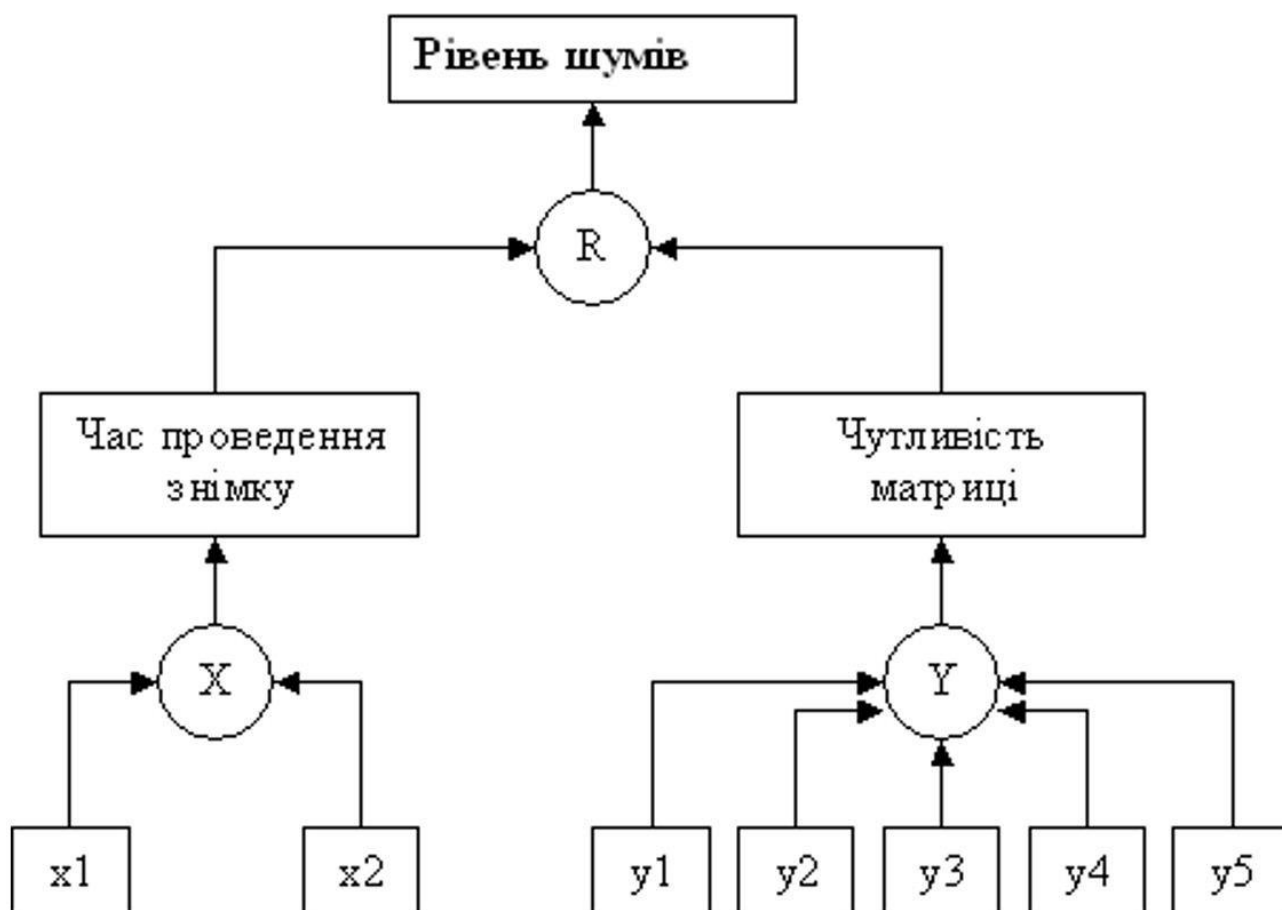


Рисунок В.5 – Схема ієрархії прийнятих показників у вигляді дерев виводу

Назва цифрової фотокамери:

Sony S7

Оцінка за рівнем шумів:

10


 OK

Рисунок В.6 – Результати роботи програми

## Додаток Г (довідниковий)

### Інструкція користувача

1). На рисунку Г.1 можна побачити початкову сторінку (головне вікно програми), у якій ми зможемо провести пробне тестування програмного продукту.

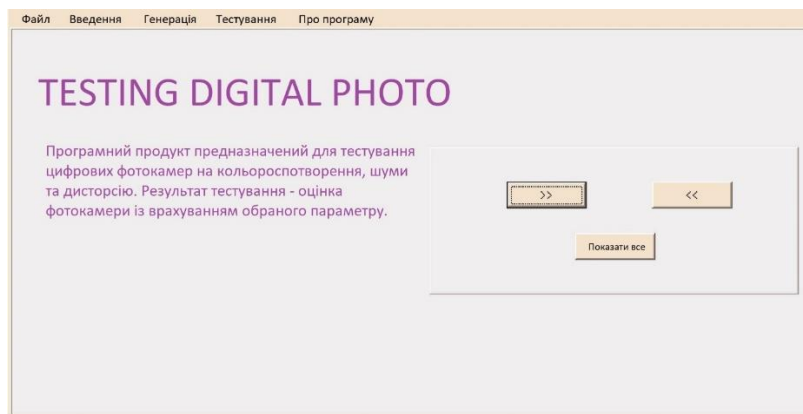


Рисунок Г.1 – головне вікно програми

2). Далі нас цікавить розділ меню, який називається тестування. При натисканні ми потрапляємо до форми, де нам потрібно заповнити усі необхідні параметри та завантажити цифрове зображення, зроблене певною фотокамерою.

3). Після того як програма обробить усі дані, можна буде дізнатися результат – оцінку фотокамери із врахування обраного параметру, в нашому випадку – рівня шуму. Результат вказаний на рисунку Г.2.

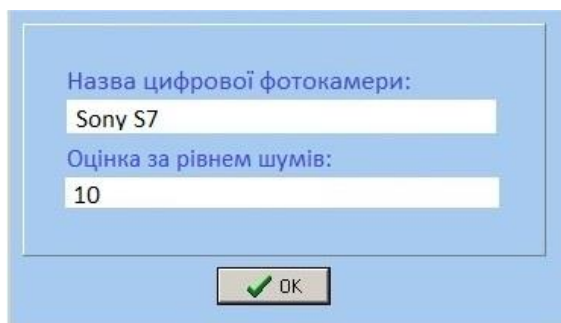


Рисунок Г.2 – результат роботи програми.