

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ,
МОЛОДІ ТА СПОРТУ УКРАЇНИ**

**ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені БОГДАНА ХМЕЛЬНИЦЬКОГО**

**ВІДДІЛЕННЯ ГІБРИДНИХ МОДЕЛЮЮЧИХ І
УПРАВЛЯЮЧИХ СИСТЕМ В ЕНЕРГЕТИЦІ
ІНСТИТУТУ ПРОБЛЕМ МОДЕЛЮВАННЯ В ЕНЕРГЕТИЦІ
імені Г.Є. ПУХОВА**

Парнюк А.М., Волошин М.В., Єремеев Б.М.

**МОДЕЛЮВАННЯ
ПАРАЛЕЛЬНИХ ПРОЦЕСІВ
В ІНФОРМАТИЦІ ТА МЕДИЦИНІ**

МОНОГРАФІЯ

*за науковою редакцією
проф. Кузьмука В.В. та доц. Супруненко О.О.*

Черкаси 2011

УДК 519.6, 004.423

ББК 22.19

П 11

Рецензенти:

Баранов В.Г., доктор технічних наук, професор,
Відділення гібридних моделюючих і управляючих систем
в енергетиці Інституту проблем моделювання
в енергетиці ім. Г.Є. Пухова НАН України;

Онищенко Б.О., кандидат фізико-математичних наук,
доцент кафедри програмного забезпечення автоматизованих систем
Черкаського національного університету
імені Богдана Хмельницького.

Парнюк А.М., Волошин М.В., Єремєєв Б.М.

П 11 Моделювання паралельних процесів в інформатиці та
 медицині. / За наук. ред. проф. Кузьмука В.В.,
 доц. Супруненко О.О.: Збірник наукових праць. – Черкаси: ЧНУ,
 2011. – 124 с.

ISBN 978-966-353-241-7

Монографія містить роботи аспірантів Відділення гібридних моделюючих і управляючих систем в енергетиці Інституту проблем моделювання в енергетиці ім. Г.Є. Пухова НАНУ, в яких розглядаються питання застосування паралельних обчислювальних процесів для розв'язання задач верифікації програмного забезпечення, розпізнавання інформативних ділянок в автоматизованих системах іриодіагностики, при реалізації засобів дистанційної біорезонансної діагностики і терапії.

УДК 519.6, 004.423

ББК 22.19

Рекомендовано до друку Вченою радою
Черкаського національного університету ім. Б. Хмельницького
(протокол № 2 від 15 листопада 2011 р.)

ISBN 978-966-353-241-7

© ЧНУ ім. Б. Хмельницького, 2011

© А.М. Парнюк, 2011

© М.В. Волошин, 2011

© Б.М. Єремєєв, 2011

© Відділення ГМУСЕ ІПМЕ

ім. Г.Є. Пухова, 2011

Зміст

Вступне слово	5
Розділ 1. Аналіз та верифікація паралельних програм на основі PN-моделей (Парнюк А.М.)	7
1.1. Актуальність верифікації і тестування паралельних програм	8
1.1.1. Проблеми тестування паралельних програм	8
1.1.2. Аналіз існуючих досліджень і публікацій	10
1.2. Верифікація і тестування програм	13
1.2.1. Завдання і цілі процесу верифікації	13
1.2.2. Тестування і верифікація – відмінності у поняттях	14
1.2.3. Тестування програмного коду	15
1.2.4. Автоматизація тестування	17
1.2.5. Методи тестування	19
1.2.5.1. Тестування «білого ящика» та «чорного ящика»	19
1.2.5.2. Статичне і динамічне тестування	20
1.2.5.3. Тестування моделей	21
1.3. Моделювання і верифікація паралельних програм	23
1.3.1. Формальні методи і тестування програмного забезпечення	23
1.3.2. Процес верифікації моделей програм	25
1.3.3. Основні поняття мереж Петрі	26
1.3.3.1. Загальні відомості про мережі Петрі	26
1.3.3.2. Види мереж Петрі	28
1.3.3.3. Принципи функціонування мереж Петрі	29
1.3.3.4. Аналіз мереж Петрі	31
1.3.4. Автоматична побудова моделей в термінах мереж Петрі	33
1.4. Огляд засобів моделювання паралельних програм	36
1.4.1. Тестування на основі моделей	36
1.4.2. Інші засоби тестування паралельних програм	40
Висновок	43
Список літератури	44
Раздел 2. Методы и средства моделирования и распознавания информативных участков в автоматических системах иридодиагностики с использованием параллелизма (Волошин М.В.)	48
2.1. История возникновения иридодиагностики	50
2.2. Основные положения иридодиагностики	51
2.3. Иридосхемы	52
2.4. Возможности иридодиагностики	54
2.5. Современное состояние иридодиагностики на Украине	55
2.6. Перспективы развития	56
2.7. Требования к ПО АРМ'а иридолога	58

2.7.1. Учет уровня профессиональной подготовки	58
2.7.2. База знаний	59
2.7.3. Логический вывод	60
2.7.4. Ввод-вывод информации	61
2.8. Компьютерные программы и рабочие станции для иридодиагностики	62
2.8.1. Компьютерные программы	62
2.8.2. Программа-лаборатория TILDA для научных исследований	71
2.8.3. Рабочие станции и специальное оборудование	72
2.8.4. Электронные учебно-методические пособия	84
2.8.5. Телекоммуникационные средства	86
2.9. Оборудование типового АРМ иридолога	86
2.10. Проектирование иридологического ПО	89
2.10.1. Поиск зрачка	90
2.10.2. Интегро-дифференциальный оператор Даугмана	94
2.10.3. Удаление шума	94
2.10.4. Уточнение границ	95
2.10.5. Определение иридопризнаков	97
2.10.6. Анализ плотности ириса	98
Заключение	100
Список литературы	101
Розділ 3. Моделі та засоби комп'ютерної діагностики енергетичних процесів в організмі людини (Єремєєв Б.М.)	103
3.1. Біорезонансна діагностика і терапія	103
3.1.1. Принцип та методика діагностики людини на основі БАТ і меридіанів	104
3.1.1.1. Метод сегментарної діагностики	104
3.1.1.2. Метод електрокупнктурної діагностики по Фолю	106
3.1.1.3. Діагностика за Накатані	109
3.1.1.4. Ауromетрія	112
3.1.2. Апаратне та програмне забезпечення для біорезонансної діагностики	113
3.1.3. Контроль і прогнозування змін станів пацієнта під час діагностування та біорезонансної терапії	118
3.2. Розробка вимог до програмного забезпечення для біорезонансної діагностики та терапії	118
3.2.1. Розробка загальних вимог до створення ПЗ	118
3.2.2. Мобільні технології в енергетичній медицині, перспективи та критерії використання	119
3.2.3. Дистанційна біорезонансна діагностика і терапія	120
Висновки	121
Список літератури	121
Додаток А	122
Додаток Б	123

ВСТУПНЕ СЛОВО

Сучасний розвиток інформаційних комп'ютерних технологій вимагає розробки нових програмних засобів, що здатні на якісно новому рівні забезпечувати розв'язання задач автоматизації технологічних процесів у експериментальних науково-дослідних роботах, енергетиці, медицині та інших галузях промисловості.

Задачі підвищення ефективності використання апаратних ресурсів вимагають застосування паралельних алгоритмів обробки даних та розв'язання задач управління в системах реального часу. Тому особливої актуальності в наш час набула проблема верифікації і, в цілому, валідації паралельного програмного забезпечення. Цьому важливому питанню присвячений огляд проблеми, представлений у першому розділі даного збірника. В даному розділі аспіранткою Парнюк А.М. зроблений огляд проблем тестування паралельних програм, які викликані, в основному, не детермінованим характером відпрацювання паралельного алгоритму. Розв'язання даної проблеми можливо при створенні інструментальних засобів діагностики програмного забезпечення, які базуються на моделях чи специфікаціях. Для побудови моделей паралельних програм застосовують потокові схеми, апарат класичних і часових логік, мережі-процеси, мережі Петрі та інші формалізми. Важливим завданням при виборі аналітично-візуального апарату для побудови моделей паралельних програм є можливість розгляду паралельних потоків подій в часі й потенціал для автоматизації пошуку помилок в паралельній моделі. Цим вимогам відповідає апарат мереж Петрі, який розглядається як підґрунтя для побудови автоматизованих засобів верифікації та тестування паралельних програм. У роботі також проведений аналіз програмних засобів тестування програмного забезпечення на основі моделей, в якому виділені основні здобутки та обмеження реалізованих на даний час спеціалізованих систем.

У другому розділі даного збірника наведені результати роботи по автоматизації процесу іридіодіагностики. Аспірант Волошин М.В. представив огляд існуючих засобів іридіодіагностики та виділив основні проблеми, з якими стикаються розробники іридіологічного програмного забезпечення, а також перспективні напрямки розвитку цієї тематики. Розглянуте апаратно-програмне забезпечення для іридіодіагностики дозволяє визначити необхідне обладнання (з відповідними характеристиками) та основні функції програмного забезпечення, виділити критичні реалізації цих функцій, по вдосконаленню яких потрібно проводити подальші дослідження. Однією з перспективних задач називається доповнення візуальних програмних засобів іридіодіагностики спеціалізованою експертною підсистемою. В роботі представлені вимоги до іридіологічного апаратно-

програмного комплексу, загальний алгоритм іридодіагностики з методами пошуку, розпізнавання та визначення іридоознак. Запропонований удосконалений алгоритм пошуку та уточнення границь ірису, представлений управляючою мережею Петрі.

Третій розділ присвячений огляду методів стаціонарної автоматизованої діагностики і терапії з метою їх застосування при створенні дистанційних засобів медичної діагностики і терапії організму людини. Аспірант Єремеев Б.М. представив огляд методів сегментарної, електроакопунктурної діагностики за Фолем та Накатані, аурометрії. Проаналізував необхідне апаратне і функції програмного забезпечення, сформулював вимоги до мобільних засобів діагностики та терапії.

Збірник буде корисний аспірантам та молодим вченим, які займаються питаннями розробки, реалізації та валідації паралельного програмного забезпечення, що використовує сучасні апаратні багатопроцесорні та багатомашинні засоби, а також розробляє та вдосконалює програмне забезпечення в медичній галузі.

Проф. Кузьмук В.В., доц. Супруненко О.О.

РОЗДІЛ 1. Аналіз та верифікація паралельних програм на основі PN-моделей (Парнюк А.М.)

Щоб задовольнити постійно зростаючі потреби у обчислювальних ресурсах, виробники обладнання все частіше роблять вибір на користь багатоядерних процесорів. Якщо при використанні одноядерних процесорів підвищення продуктивності досягається в основному за рахунок підвищення тактової частоти, то при використанні багатоядерних систем успіх залежить від того, наскільки ефективно реалізована паралельна обробка у додатку.

Тому особливо актуальним на даний час є завдання створення паралельного програмного забезпечення, що обумовлює виникнення нових технологій створення паралельних програм. Важливим питанням при цьому є неперервне підвищення продуктивності обчислювальних систем та ефективності використання апаратних ресурсів. При створенні працездатної паралельної програми програміст стикається з рядом проблем, які полягають у врахуванні типу пам'яті та особливостей звертання до неї, в організації взаємодії паралельних процесів, у тестуванні програмного продукту на працездатність, наявність надлишковості, неоднозначностей і тупикових ситуацій, у адаптації алгоритму до динаміки завантаження вільних обчислювальних ресурсів. У зв'язку з цим важливим завданням є створення автоматизованих програмних засобів аналізу паралельних програм.

Ця задача найбільш вдало вирішується із застосуванням модельного підходу при використанні засобів візуального представлення топології та функціонування паралельного алгоритму у сполученні з аналітичними засобами тестування досліджуваної алгоритмічної моделі. Для вирішення цього завдання шляхом візуального представлення паралельних процесів та пошуку критичних властивостей можна застосовувати апарат мереж Петрі.

Мережі Петрі являють собою графічний і математичний засіб моделювання асинхронних паралельних процесів. Вони є перспективним візуальним інструментом опису й дослідження мультипрограмних, асинхронних, розподілених, паралельних, не детермінованих і/або стохастичних систем обробки інформації. Як графічний засіб мережі Петрі можуть використовуватися для наочного подання модельованої системи, подібно граф-схемам алгоритмів та мережним графікам. Поняття фішок (міток), що вводиться в цих мережах, дозволяє відображати структуру та моделювати динаміку функціонування паралельних процесів. Як математичний засіб аналітичне подання мережі Петрі дозволяє створювати математичне відображення динаміки функціонування модельованої системи. Мережі Петрі можуть із успіхом використовуватися й теоретиками, і практиками, а, отже, стають ефективним засобом їхнього взаємного спілкування: практики можуть перейняти в теоретиків методологію

побудови моделей, а теоретики – навчитися в практиків, як наблизити свої моделі до конкретної реалізації.

1.1. Актуальність верифікації і тестування паралельних програм

Паралельна обробка інформації широко використовується для збільшення продуктивності обчислювальних систем. Особливе місце серед паралельних систем займають системи реального часу. Процес тестування систем такого типу - нетривіальна задача, що вимагає для свого рішення фундаментальних досліджень, що базуються на різних формальних методах і засобах, які варіюються залежно від класу модельованих систем, ступеня деталізації їх структури і поведінки, а також від характеру досліджуваних проблем.

1.1.1 Проблеми тестування паралельних програм

Тестування паралельних додатків – завдання непросте. Помилки розпаралелювання складно виявити через недетермінованість поведінки паралельних програм. Навіть якщо помилку виявлено, її часто складно відтворити повторно. Крім того, після модифікації коду, не так просто переконатися, що помилка дійсно усунена, а не замаскована. Можна сказати, що помилки в паралельній програмі є класичними "гейзенбагами" (термін, використовуваний у програмуванні для опису програмної помилки, яка зникає або змінює свої властивості при спробі її виявлення, у російській термінології більш часто використовується термін "плаваюча помилка"). Прикладом можуть бути помилки, які виявляються в остаточному варіанті програми ("релізі"), проте їх не видно в режимі налагодження, або помилки синхронізації в багатопотоковому додатку [1].

До того ж проблема полягає ще й в тому, що паралельне програмування набагато важче послідовного програмування. Врахувати всі аспекти паралелізму дуже важко навіть при багатопотоковому програмуванні з використанням розподіленої пам'яті (хоча в цьому підході частково зберігається стиль послідовного програмування) тому, що можливо використовувати чергування базових операцій. Неврахування будь-яких аспектів може призвести до помилок («гонки даних», data race) або незавершеності програми (синхронізаційний тупик, deadlock). Такі дефекти можуть не виявлятися при тестуванні програмного забезпечення і приводити до катастрофічних результатів під час його експлуатації [2].

Таким чином, завдання паралельного тестування багато в чому зводиться до проблеми створення інструментів діагностики, що мінімально впливають на поведінку програми або створюють необхідні умови для її прояву [1].

Рішення перерахованих проблем пропонує нове покоління інструментів, які базуються на підході тестування на основі моделі (model based testing) або на основі специфікацій (specification based testing).

Тому не дивно, що сучасний етап розвитку теоретичної інформатики характеризується бурхливим зростанням активності досліджень в області розроб-

ки формальних методів специфікації, аналізу і моделювання паралельних/розподілених систем, які мають складну структурну організацію і функціонують в режимі реального часу. Розробка коректних систем такого типу – нетривіальна задача, що вимагає для свого успішного вирішення проведення комплексних фундаментальних досліджень, що базуються на різних формальних методах і засобах, які варіюються залежно від класу модельованих систем, ступеня деталізації їх структури і поведінки, а також характеру досліджуваних проблем. На основі результатів і рекомендацій теоретичних досліджень ведеться пошук і перевірка нових архітектурних принципів конструювання паралельних/розподілених систем, вивчаються методи розпаралелювання алгоритмів і програм, перевіряються нові способи організації програм і процесів, обґрунтовуються програмні конструкції, що вводяться в мови паралельного програмування, відпрацьовуються методи структурного і семантичного аналізу паралельних програм і т.д.

Протягом трьох останніх десятиліть теорія паралелізму породила велику різноманітність моделей, теорем, алгоритмів та інструментів, призначених для специфікації, розробки та верифікації паралельних/розподілених систем [3]. При цьому використовуються методи і поняття теорії графів, теорії множин, теорії алгоритмів, теорії категорій, математичної логіки та лінійне програмування. В якості формальних моделей паралелізму застосовують різні класи та узагальнення структур подій, мереж Петрі, мереж-процесів, потокових схем, апарат тем поральних логік [4].

Вивчаються фундаментальні поняття і закони паралельної обробки інформації і на підставі виявлених закономірностей будуються деталізовані формальні моделі досліджуваних об'єктів, на яких ставляться і вирішуються прикладні завдання. Дослідження ведуться у трьох основних напрямках: формальні моделі, алгебра процесів, логіка процесів.

Таким чином, можна констатувати, що до цього моменту вже склався певний, умовно кажучи, 'класичний' підхід до розробки коректних паралельних систем, який, як добре відомо, має ряд обмежень: існує можливість дослідження тільки систем з простою структурною організацією і кінцевим числом станів; не до кінця вивчені взаємозв'язки між базовими відношеннями подій паралельних систем; детально опрацьована тільки інтерлівінгова семантика паралельних процесів; відсутні еквівалентні поняття, що відображають базові відношення подій розподілених систем, не встановлено взаємозв'язки між різними моделями та підходами (наприклад, семантичними, алгебраїчними і логічними); ефективні верифікаційні алгоритми (алгоритми перевірки на моделях) розроблені тільки для темпоральної логіки CTL; має місце значне зниження ефективності верифікації через проблеми 'вибуху станів'; недостатньо пророблені часові аспекти функціонування паралельних систем і т.і. [3].

1.1.2 Аналіз існуючих досліджень і публікацій

Останнім часом все більш широке поширення в усьому світі одержують супер-комп'ютери, а елементи паралелізму потроху проникають в усі типи комп'ютерів аж до персональних. Тому завдання створення ефективних паралельних програм, а також адаптація для паралельних комп'ютерів величезної кількості вже існуючих послідовних додатків стає усе більше важливим.

У розробку теорії, що лежить в основі створення систем статичного аналізу інформаційної структури програм, внесли свій внесок багато вчених в усьому світі, такі як В.А. Серебряков, В.Н. Касьянов [5], А.А. Летичевский, В.В. Воєводін [6], М. Wolfe, W. Pugh, K. Kennedy, M. Lam, F. Irigoín, P. Tang, A. Schouten, L. Lamport, D. Kuck, P. Feautrier, U. Banerjee та інші [7],[8],[9].

Особливість паралельних програм у порівнянні з послідовними програмами в тому, що в загальному випадку виконання паралельної програми є недетермінованим, тому що залежить від багатьох факторів, що впливають на виконання кожного процесу. Тому для доказу працездатності паралельної програми недостатньо набору тестів, що покривають область значень вхідних даних. Необхідно також довести, що, або поведінка програми є детермінованою, або результат роботи паралельної програми не залежить від умов виконання. Таким чином, хоча рішення проблеми налагодження може спростити процес розробки паралельних програм, доказ працездатності паралельних програм вимагає залучення математичних точних методів перевірки коректності програми або, принаймні, детермінованості її поведінки.

У загальному випадку питання коректності програм розбивається на дві частини. По-перше, під коректністю програми мається на увазі відсутність у ній граматичних, синтаксичних помилок, помилок перетворення типів даних і т.п. й, відповідно, працездатність програми. По-друге, під коректністю програми також мається на увазі виконання програмою поставленою перед нею задачі, що можна визначити як коректність вибору й реалізації алгоритму.

Автоматизувати процес доказу коректності вибору алгоритму практично неможливо, тому що для цього потрібно формалізувати задачу, розв'язвану алгоритмом у термінах, що розуміються автоматичним аналізатором, а це в загальному випадку приводить до необхідності написання подібного ж алгоритму [65], але в інших термінах, коректність якого також необхідно доводити. Таким чином, питання коректності програми необхідно звзвити до працездатності програми й коректності реалізації алгоритму.

Доказ працездатності програм називається верифікацією [10]. Методи й засоби верифікації програм розробляються протягом уже багатьох десятиліть. Незважаючи на те, що вже існують засоби верифікації більш-менш складних програм, автоматична верифікація в цей час не є обов'язковою частиною процесу розробки програм. Для паралельних же програм, налагодження яких ускладнене внаслідок багатоваріантності виконання, верифікація програм стає особливо важливою.

Коректність реалізації алгоритму в послідовному випадку звичайно розглядається як задачу Model Checking [8], де специфікація завдання описується формулами темпоральної логіки, а програма представляється у вигляді деякої системи переходів, або як задачу Theorem Proving, де за допомогою автоматизованих процедур формального висновку доводиться, що результат обчислень програми відповідає поставленому завданню. У паралельному програмуванні крім наведених властивостей коректності програми існує також необхідність доказу коректності роботи програми, зокрема, відсутність у паралельній програмі дедлоків, досяжність кінцевого стану, а при наданні паралельною програмою інтерактивного сервісу іншим програмам, до питань коректності також належить відсутність непродуктивних циклів і живість паралельної програми (мережі).

Серед вітчизняних досліджень в області розробки формальних методів аналізу паралельних систем і процесів слід відзначити роботи Н.А. Анісімова, О.Л. Бандман, В.А. Вальковського [7], Ю.Г. Карпова, В.Є. Котова [11], Р.М. Смілянського, В.А. Соколова [9], Л.А. Черкасової.

При перевірці коректності роботи паралельних програм крім тестування й відлагодження використовують різні техніки верифікації, серед яких однією з основних є техніка модельного підходу [6-7]. Ця техніка верифікації ґрунтується на аналізі властивостей моделей програм, виражених у термінах різних математичних теорій. Такі теорії надають точні способи й методи для верифікації, що дозволяють формально проаналізувати ті властивості паралельних програм, які нас цікавлять, наприклад, виявляти помилки, що виникають при організації взаємодії процесів. У рамках даної техніки верифікації для створення моделей використовують поняття з теорій автоматів, графів і мереж Петрі, а також поточкові діаграми й машини Тюрінга [7]. При цьому важливу роль відіграє візуальне подання даних моделей.

Формальний апарат мереж Петрі призначений для моделювання й аналізу розподілених систем [12, 13]. За час свого розвитку він накопичив безліч методів і алгоритмів аналізу. Значна частина цих методів і алгоритмів заснована на побудові дерева досяжності або іншими словами простору станів розглянутої моделі. Ці методи успішно застосовуються для аналізу моделей, що створюються фахівцями з мереж Петрі, і відмінною рисою таких моделей є порівняно невеликий простір станів. Спроба побудувати повний простір стану для паралельної програми приречена на невдачу, тому що кількість станів паралельної програми легко може перевищити оперативну пам'ять будь-якого суперкомп'ютера. Тому необхідно шукати способи зменшення простору станів паралельної програми. Найпростіший спосіб скорочення простору станів в даному випадку це не використовувати інформацію про змінні в моделі.

Візуалізація програмного забезпечення паралельних обчислень містить у собі дослідження й розробку візуальних мов паралельного програмування, візуальних відладчиків правильності, а також систем налагодження, вимірювання й аналізу

продуктивності паралельних програм. Якщо у випадку послідовних систем візуалізація програмного забезпечення (Software Visualization) грає найчастіше лише допоміжну роль, то в паралельних обчисленнях вона, як правило, виявляється життєво необхідною.

Насамперед це пов'язане з відсутністю систем паралельного програмування, що дозволяють ефективно автоматизувати процеси програмування й налагодження у випадку паралельних обчислень і тим самим понизити їхню інтелектуальну складність. Візуалізація служить засобом полегшення тих складностей, які несе паралельне програмування, і особливо відлагодження паралельних програм.

Друге завдання візуалізації – забезпечення ефективності (кращої продуктивності) паралельних програм. Тому значна частина проектів, що вийшли за рамки академічних, належать до систем налагодження, настроювання, виміру й аналізу продуктивності паралельних програм. Більшість цих систем забезпечує збір даних і відображення метрик продуктивності паралельних програм. При цьому, основні проблеми виникають не стільки в процесі пошуку причин неефективності, скільки при з'ясуванні того, що потрібно зробити, щоб програма працювала з належним прискоренням.

В області побудови й аналізу моделей готових паралельних програм одним з ефективних і наочних математичних формалізмів є мережі Петрі. Тут можна згадати роботи, виконані під керівництвом В. А. Соколова [9] з побудови моделей паралельних програм, запрограмованих мовою Сі, а також роботи з побудови моделей паралельних програм, запрограмованих мовою Паскаль. Із закордонних робіт можна згадати методи трансляції програм мовою Ада з використанням мереж Петрі [14, 15].

Проте слід зазначити, що незважаючи на перспективність побудови й аналізу моделей готових паралельних програм, помітний розрив між теоретичною базою й існуючими автоматичними засобами побудови й аналізу моделей.

1.2. Верифікація і тестування програм

1.2.1 Завдання та цілі процесу верифікації

Спочатку розглянемо цілі верифікації. Основна мета даного процесу – доказ того, що результат розробки відповідає пред'явленим до нього вимогам. Зазвичай процес верифікації проводиться зверху вниз, починаючи від загальних вимог, заданих в технічному завданні і/або специфікації на всю інформаційну систему до детальних вимог на програмні модулі і їхню взаємодію. До складу завдань процесу входить послідовна перевірка того, що в програмній системі:

- загальні вимоги до інформаційної системи, призначені для програмної реалізації, коректно перероблені в специфікацію вимог високого рівня до комплексу програм, що задовольняють вихідним системним вимогам;
- вимоги високого рівня правильно перероблені в архітектуру ПЗ та в специфікації вимог до функціональних компонентів низького рівня, які задовольняють вимогам високого рівня;
- специфікації вимог до функціональних компонентів ПЗ, розташованим між компонентами високого і низького рівня, задовольняють вимогам більш високого рівня;
- архітектура ПЗ та вимоги до компонентів низького рівня коректно перероблені в задовольняючі їм вихідні тексти програмних та інформаційних модулів;
- вихідні тексти програм і відповідний їм виконуваний код не містять помилок.

Крім того, для верифікації на відповідність специфікації необхідні вимоги на конкретний проект програмного засобу, вимоги до технологічного забезпечення життєвого циклу ПЗ, а також вимоги до експлуатаційної та технологічної документації.

Цілі верифікації ПЗ досягаються за допомогою послідовного виконання комбінації з інспекцій проектної документації та аналізу їх результатів, розробки тестових планів тестування та тест-вимог, тестових сценаріїв і процедур та подальшого виконання цих процедур. Тестові сценарії призначені для перевірки внутрішньої несуперечності та повноти реалізації вимог.

Виконання тестових процедур повинне забезпечувати демонстрацію відповідності випробовуваних програм вихідним вимогам.

На вибір ефективних методів верифікації та послідовність їх застосування в найбільшій мірою впливають основні характеристики тестованих об'єктів:

- клас комплексу програм, визначається глибиною зв'язку його функціонування з реальним часом та випадковими впливами з зовнішнього середовища, а також вимоги до якості обробки інформації та надійності функціонування;
- складність або масштаб (об'єм, розміри) комплексу програм і його функціональних компонентів, що є кінцевими результатами розробки;

- переважні елементи в програмах: здійснюють обчислення складних виразів та перетворення вимірюваних величин або обробні логічні і символічні дані для підготовки та відображення рішень.

Визначимо деякі поняття та визначення, пов'язані з процесом тестування, як складової частини верифікації. Майерс дає наступні визначення:

Тестування – це процес виконання програми з метою виявлення помилки [16, 17].

Таким чином, в процесі тестування програмного забезпечення, як правило, перевіряють наступне:

- відповідність програмного забезпечення вимогам щодо його функціонування,
- адекватність функціонування програмного забезпечення в ситуаціях, не відображені у вимогах, тобто перевірка системи на відмови.
- відсутність у програмному забезпеченні типових помилок, які роблять програмісти [10].

1.2.2 Тестування та верифікація – відмінності у поняттях

Незважаючи на схожість, терміни «тестування» та «верифікація» позначають різні рівні перевірки коректності роботи програмної системи. Щоб уникнути подальшої плутанини, чітко визначимо ці поняття. Тестування програмного забезпечення – вид діяльності в процесі розробки, пов'язаний з виконанням процедур, спрямованих на виявлення (доказ наявності) помилок (невідповідностей, неповноти, двозначностей і т.д.) в поточному визначенні розроблюваної програмної системи. Процес тестування належить в першу чергу до перевірки коректності програмної реалізації системи, відповідності вимогам реалізації, тобто тестування – це кероване виконання програми з метою виявлення невідповідностей її поведінки вимогам.

Верифікація програмного забезпечення – більш загальне поняття, ніж тестування. Метою верифікації є досягнення гарантії того, що верифікований об'єкт (вимоги або програмний код) відповідає вимогам, реалізований без непередбачуваних функцій і задовольняє проектним специфікаціям і стандартам. Процес верифікації включає в себе інспекції, тестування коду, аналіз результатів тестування, формування і аналіз звітів про проблеми. Таким чином, прийнято вважати, що процес тестування є складовою частиною процесу верифікації (рис.1).

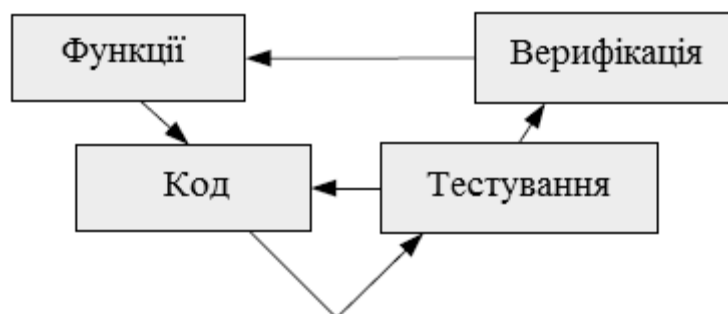


Рис. 1. Тестування і верифікація.

Якщо подивитися на ці процеси з точки зору питання, на яке вони дають відповідь, то тестування відповідає на питання «Як це зроблено?» або «чи відповідає поведінка розробленої програми вимогам?», а верифікація - «Що зроблено?» або «Чи відповідає розроблена система вимогам?» [10].

1.2.3 Тестування програмного коду

Тестування програмного коду – процес виконання програмного коду, направлений на виявлення існуючих в ньому дефектів. Під дефектом тут розуміється ділянка програмного коду, виконання якого за певних умов призводить до несподіваної поведінки системи (тобто поведінки, що не відповідає вимогам). Несподівана поведінка системи може призводити до збоїв у її роботі і відмов, в цьому випадку говорять про істотні дефекти програмного коду. Деякі дефекти викликають незначні проблеми, що не порушують процес функціонування системи, але дещо ускладнюють роботу з нею. У цьому випадку говорять про середні або малозначні дефекти.

Завдання тестування при такому підході – визначення умов, за яких проявляються дефекти системи і протоколювання цих умов. У завдання тестування зазвичай не входить виявлення конкретних дефектних ділянок програмного коду і ніколи не входить виправлення дефектів – це завдання налагодження, що виконується за результатами тестування системи.

Мета застосування процедури тестування програмного коду – мінімізація кількості дефектів, особливо істотних, в кінцевому продукті. Тестування саме по собі не може гарантувати повної відсутності дефектів у програмному коді системи. Однак, у поєднанні з процесами верифікації та валідації, спрямованими на усунення суперечливості та неповноти проектної документації (зокрема - вимог на систему), грамотно організоване тестування дає гарантію того, що система задовольняє вимогам і веде себе відповідно до них у всіх передбачених ситуаціях.

При розробці систем підвищеної надійності, наприклад, авіаційних, гарантії надійності досягаються за допомогою чіткої організації процесу тестування, визначення його зв'язку з іншими процесами життєвого циклу, введення кількісних характеристик, що дозволяють оцінювати успішність тестування. При цьому, чим вище вимоги до надійності системи (її рівень критичності), тим жорсткіші вимоги пред'являються.

Таким чином, в першу чергу ми розглядаємо не конкретні результати тестування конкретної системи, а загальну організацію процесу тестування, використовуючи підхід «добре організований процес дає якісний результат». Такий підхід є загальним для багатьох міжнародних і галузевих стандартах якості. Якість розроблюваної системи при такому підході є наслідком організованого процесу розробки та тестування, а не самостійним некерованим результатом [17].

Оскільки сучасні програмні системи мають досить значні розміри, при тестуванні їх програмного коду використовується метод функціональної

декомпозиції. Система розбивається на окремі модулі (класи, простори імен і т.і.), що мають відповідні вимогам функціональність та інтерфейси. Після цього окремо тестується кожен модуль – виконується модульне тестування.

Потім виконується складання окремих модулів в більш великі конфігурації – виконується інтеграційне тестування, і нарешті, тестується система в цілому – виконується системне тестування.

У ході модульного тестування кожен модуль тестується як на відповідність вимогам, так і на відсутність проблемних ділянок програмного коду, які можуть викликати відмови і збої в роботі системи. Як правило, модулі не працюють поза системою – вони приймають дані від інших модулів, переробляють їх і передають далі. Для того, щоб з одного боку, ізолювати модуль від системи і виключити вплив потенційних помилок системи, а з іншого боку – забезпечити модуль усіма необхідними даними, використовується тестове оточення.

Завдання тестового оточення – створити середовище виконання для модуля, емулювати всі зовнішні інтерфейси, до яких звертається модуль.

Типова процедура тестування полягає в підготовці і виконанні тестових прикладів (також вони називаються просто тестами). Кожен тестовий приклад перевіряє одну «ситуацію» в поведінці модуля і складається зі списку значень, що передаються на вхід модуля, опису запуску та виконання обробки даних – тестового сценарію, і списку значень, які очікуються на виході модуля в разі його коректної поведінки. Тестові сценарії складаються таким чином, щоб виключити звернення до внутрішніх даних модуля, всі взаємодії мають відбуватися лише через його зовнішні інтерфейси.

Виконання тестового прикладу підтримується тестовим оточенням, яке включає в себе програмну реалізацію тестового сценарію. Виконання починається з передачі модулю вхідних даних та запуску сценарію. Реальні вихідні дані, отримані від модуля в результаті виконання сценарію зберігаються і порівнюються з очікуваними. У разі їх збігу тест вважається пройденим, в іншому випадку – не пройденим. Кожен не пройдений тест вказує або на дефект в тестованому модулі, або в тестовому оточенні, або в описі тесту.

Сукупність описів тестових прикладів складає тест-план – основний документ, що визначає процедуру тестування програмного модуля. Тест-план задає не тільки самі тестові приклади, але й порядок їх проведення, який також може бути важливим.

Під час тестування часто буває необхідно враховувати не тільки вимоги до системи, але і структуру програмного коду тестованого модуля. У цьому випадку тести складаються таким чином, щоб детектувати типові помилки програмістів, викликані неправильною інтерпретацією вимог. Застосовуються перевірки граничних умов, перевірки класів еквівалентності. При відсутності в системі можливостей, які не задані вимогами, гарантують різні оцінки покриття програмного коду тестами, тобто оцінки того, який відсоток тих чи інших мовних конструкцій виконаний в результаті виконання всіх тестових прикладів [10].

1.2.4 Автоматизація тестування

В даний час промислове виробництво програмного забезпечення (ПЗ) досягло таких масштабів і такого рівня складності, що необхідність в індустріальних технологіях систематичного тестування загальноновизнана. Особливо актуальним є створення таких технологій, які забезпечують одночасно якісне, систематичне тестування цільового ПЗ і високий ступінь автоматизації розробки тестів [18]. Традиційні методи розробки тестів вручну вже не можуть забезпечити якісне тестування сучасних програмних систем [19]. А отже автоматизація тестування може дати можливість виконувати певні тестові завдання набагато швидше і ефективніше ніж це може бути зроблено вручну. Переваги даного підходу наведені нижче.

1. Виконання існуючих регресійних тестів. Це має бути найочевидніше завдання автоматизації, зокрема, коли програми виправляються і доповнюються дуже часто. Зусилля, витрачені на виконання існуючих регресійних тестів повинні бути мінімальні, для того, щоб у потрібний момент була можливість вибору автоматизованих тестів і їх запуск з мінімальними ручними зусиллями і за лічені хвилини.

2. Виконання великої кількості тестів більш часто. Відомий факт, що часто-повторювані завдання найбільш добре піддаються автоматизації. Чітка вигода в тому, щоб запускати більшу кількість тестів, за менші терміни, даючи при цьому непряму можливість запускати тести більш часто.

3. Виконання тестів, які складно або неможливо розробити. Припустимо користувач, дивлячись на графічний інтерфейс, інтуїтивно очікує появу якоїсь візуалізації даних при певних діях, але користувач (тестувальник) може навіть і не уявляти, що за цим криються якісь приховані механізми, такі як, наприклад, виклик і обробка подій, які не можна перевірити візуально. Події та їх поведінку просто неможливо перевірити без використання спеціальних програмних засобів.

4. Використовувати зекономлені ресурси на більш важливі й цікаві завдання. Автоматизація звичайних і нудних завдань, наприклад, повторюваних з різними варіаціями даних, дає можливість покладатися на велику ступінь акуратності при проведенні тестування і при цьому поліпшується моральний стан людей, які працюють з такими завданнями. Також висококваліфіковані працівники можуть вільно вкладати свої зусилля в проектування більш складних автоматичних тестів, які згодом будуть виконуватися. Ручна робота нікуди не дінеться. Ручне тестування є невід'ємною частиною автоматизованого тестування і навпаки. Завжди будуть завдання, які краще виконувати вручну, більш того ручне тестування буде виконуватися набагато краще, якщо автоматичні тести нададуть впевненість у тому, що в протестованому ділянці немає дефектів.

5. Постійність і повторюваність тестів. Автоматизація дає нам можливість постійно виконувати тести з чітко визначеними одними і тими ж даними (вхідні дані завжди будуть ті ж, а от вихідні можуть відрізнитися протягом деякого часу розробки), чого нам на жаль не може гарантувати людина, виконуючи цю ж

роботу вручну, тому впевненість у тестуванні підвищується. Ці ж тести можуть бути виконані на різних операційних системах і з використанням різних баз даних, що теж дасть впевненість в тому, що кросплатформенність продукту на високому рівні.

6. Повторне використання тестів. Зусилля, вкладені на прийняття рішень, що необхідно тестувати, проектування та формування тестів може дати можливість використовувати або повторно використовувати розроблені тести. У свою чергу це заощадить час розробки нових тестів. Необхідно враховувати, що б такі тести були стабільними, інакше витрачений час на підтримку тестів буде більшим.

7. Швидкий вивід продукту на ринок. Як тільки тести будуть автоматизовані, у вас з'явиться можливість повторного запуску, а загальний час, витрачений на виконання тестування, буде меншим.

8. Підвищується впевненість. Звичайно ж, усвідомлення і розуміння, що вичерпна кількість автоматичних тестів дає постійно позитивні результати збільшує довіру і впевненість у тому, що не буде ніяких непередбачуваних сюрпризів, коли програмне забезпечення буде випущено (звичайно, мається на увазі, що тести виконують свою функції і перевіряють ПЗ на наявність помилок).

Якщо скласти все вищесказане, то більший обсяг тестування, може бути досягнуто з меншими зусиллями, даючи при цьому можливість покращувати як якість, так і продуктивність [18].

Зазвичай автоматизація тестування зводиться до автоматизації виконання тестів і генерації звітів за їх результатами. Автоматизувати підготовку тестів і аналіз отриманих результатів важче, оскільки при цьому необхідно звернення до вимог до ПЗ, відповідність яким повинна бути підтверджена під час тестування. Вимоги ж часто представлені у вигляді неформальних документів, а іноді - тільки як знання та досвід експертів, аналітиків і проектувальників ПЗ. Для того, щоб залучити вимоги в автоматизований процес розробки тестів, необхідно перевести їх у формальне подання, яке може бути оброблене повністю автоматично. Для цієї мети вимоги описують у вигляді формальних специфікацій цільової системи, які можна перетворити на програми, що виконують перевірку відповідності роботи цільового ПЗ зафіксованим в них вимогам.

Не дивлячись на активний розвиток методів побудови тестів на основі формальних специфікацій або формальних моделей, лише деякі з них можуть застосовуватись в індустрії виробництва ПЗ. Основна проблема тут у тому, що індустрії потрібні не окремі методи, а технології, тобто інструментальні системи методів для вирішення наборів задач, які належать до виділеного аспекту розробки ПЗ [19].

1.2.5 Методи тестування

1.2.5.1 Тестування «білого ящика» і «чорного ящика»

1.2.5.1.1 Тестування «Чорного ящика» [10], [20], [21].

Основна ідея у тестуванні системи, як чорного ящика полягає в тому, що всі матеріали, які доступні тестувальнику – вимоги на систему, що описують її поведінку і сама система, працювати з якою він може тільки подаючи на її входи деякі зовнішні впливи і спостерігаючи на виходах деякий результат. Всі внутрішні особливості реалізації системи приховані від тестувальника, таким чином, система і являє собою «чорний ящик», правильність поведінки якого щодо до програмних вимог і належить перевірити.

З точки зору програмного коду чорний ящик може представляти собою набір класів (або модулів) з відомими зовнішніми інтерфейсами, але недоступними вихідними текстами.

Основне завдання тестувальника для даного методу тестування полягає в послідовній перевірці відповідності поведінки системи вимогам. Крім того, тестувальник повинен перевірити роботу системи в критичних ситуаціях – що відбувається у разі подання невірних вхідних значень. В ідеальній ситуації всі варіанти критичних ситуацій повинні бути описані у вимогах на систему і тестувальнику залишається тільки реалізовувати конкретні перевірки цих вимог. Проте в реальності в результаті тестування зазвичай виявляється два типи проблем системи:

1. Невідповідність поведінки системи вимогам.
2. Неадекватна поведінка системи в ситуаціях, не передбачених вимогами.

Звіти про обидва типи проблем документуються та передаються розробникам. При цьому проблеми першого типу зазвичай викликають зміну програмного коду, набагато рідше – зміну вимог. Зміна вимог в даному випадку може знадобитися зважаючи на їх суперечливості (кілька різних вимог описують різні моделі поведінки системи в одній і тій же самій ситуації) або некоректність (вимоги не відповідають дійсності).

Проблеми другого типу однозначно вимагають зміни вимог з огляду на їх неповноту – у вимогах явно пропущена ситуація, що призводить до неадекватного поведінки системи. При цьому під неадекватною поведінкою може розумітися як повний крах системи, так і взагалі будь-яка поведінка, що не описана в вимогах.

Тестування чорного ящика називають також тестуванням з вимогами, так як це єдине джерело інформації для побудови тест-плану.

1.2.5.1.2 Тестування «Білого ящика» [10], [20], [22].

Під час тестування системи, як білого ящика, тестувальник має доступ не тільки до вимог на систему, її входів і виходів, а й до її внутрішньої структури – бачить її програмний код.

Доступність програмного коду розширює можливості тестувальника тим, що він може бачити відповідність між вимогами і ділянками програмного коду і тим самим визначити – чи на весь програмний код існують вимоги. Програмний код, для якого відсутні вимоги називають кодом, непокритим вимогами. Такий код є потенційним джерелом неадекватної поведінки системи. Крім того, прозорість системи дозволяє поглибити аналіз її ділянок, які викликають проблеми – часто одна проблема нейтралізує іншу і вони ніколи не виникають одночасно.

Під час тестування сірого ящика розробник тесту має доступ до вихідного коду, але при безпосередньому виконанні тестів доступ до коду, як правило, не потрібний.

1.2.5.2 Статичне і динамічне тестування

Описані вище техніки – тестування білого ящика і тестування чорного ящика – припускають, що код виконується, і різниця полягає лише в тій інформації, якою володіє тестувальник. В обох випадках це динамічне тестування.

При статичному тестуванні програмний код не виконується – аналіз програми відбувається на основі початкового коду, який вичитується вручну, або аналізується спеціальними інструментами. У деяких випадках, аналізується не вихідний, а проміжний код (такий як байт-код або код на MSIL).

1.2.5.2.1 Статичний аналіз коду [23], [24].

Статичним є аналіз коду без запуску програми. Зазвичай при статичному аналізі перевіряються метадані скомпільованої програми або анотованого вихідного коду. Найчастіше в нього також включається якийсь етап формальної перевірки, який повинен гарантувати, що припущення, на які спирався програміст під час розробки, не призводять до неправильного поведінки програми. З усіх засобів статичного аналізу некерованих додатків найбільш популярні Prefix і Prefast. Для керованих додатків найчастіше застосовується FxCop.

Статичний аналіз має як переваги, так і недоліки. В якості переваг можна відзначити детальність і повноту охоплення аналізованого коду, можливість переконатися в правильності архітектури програми, точність звітів про помилки, що спрощують пошук і усунення дефектів у програмі. З іншого боку, статичний аналіз дійсно допомагає усунувати дефекти тільки при наявності достатньої кількості коментарів до коду. До того ж коментарі мають бути правильними. Крім того, засоби статичного аналізу дають велику кількість помилкових позитивних результатів і вимагають чималих зусиль для їх мінімізації.

1.2.5.2 Динамічний аналіз коду [23], [25].

При динамічному аналізі відстеження дефектів ведеться за результатами запуску програми. Динамічний аналіз буває двох типів: інтерактивний і автономний. Засоби інтерактивного динамічного аналізу перевіряють додаток безпосередньо під час його роботи; засоби автономного динамічного аналізу записують трасувальні дані, а потім перевіряють записи, виявляючи дефекти. Динамічний аналіз зручний тим, що не вимагає практично ніяких зусиль від розробника на етапі створення коду, він дає менше помилкових позитивних результатів і спрощує усунення найбільш очевидних проблем. Оскільки дефекти програми визначаються тільки при виконанні програми, першими проявляються проблеми, пов'язані з найбільш часто виконуваними операціями. Така особливість динамічного аналізу знижує витрати на підвищення стабільності програми.

Але й недоліки у динамічного аналізу теж є. Перевірка відбувається виключно під час роботи програми, і відсоток охоплення коду при тестуванні залежить від того, наскільки добре написані тести. Насправді деякі засоби здатні зафіксувати гонки, тільки якщо вони були в даному сеансі роботи програми. Тобто, якщо засіб аналізу повідомляє, що помилок немає, все одно не можна остаточно бути впевненим в цьому. Ще один недолік полягає у тому, що більшість засобів динамічного аналізу залежить від оснащення програми засобами моніторингу, що може змінювати поведінку виконуючого середовища. Через складність продуктивність таких засобів не надто висока.

1.2.5.3 Тестування моделей

Тестування моделей знаходиться трохи осторонь від класичних методів верифікації програмного забезпечення. Перш за все це пов'язано з тим, що об'єкт тестування – не сама система, а її модель, спроектована формальними засобами [10].

Model checking – це один з найбільш перспективних і широко використовуваних підходів до вирішення проблеми автоматизації налагодження та перевірки правильності програм. Його суть проста. Для заданої аналізованої програми будується її абстрактна формальна модель (наприклад, система переходів). Перевіряється, чи виражається властивість або вимога на формальною математичною мовою (наприклад, представляється у вигляді логічної формули).

Після цього верифікація програми зводиться до перевірки виконаності формалізованої вимоги (специфікації) на абстрактній моделі програм. Як тільки вдається знайти алгоритм побудови моделі за заданою програмою і алгоритм перевірки специфікації на заданій моделі, ми отримуємо можливість побудувати автоматичну систему верифікації програм, зручну для практичного застосування. Все, що потрібно від користувача такої системи, – це описати властивість програми, яку необхідно перевірити, на відповідній формальній мові специфікації [8].

Цей метод допускає формальний дедуктивний аналіз. Засіб перевірки моделей намагається змоделювати стан або взаємоблокування. Перевірка на основі моделей дозволяє формально обґрунтувати відсутність цих дефектів. Такий метод допомагає переконатися в правильності проекту та архітектури програми, повністю охоплює весь його код і вимагає установки мінімального числа зовнішніх драйверів [23].

Якщо залишити осторонь питання перевірки коректності та застосовності самої моделі (вважається, що її коректність і відповідність вихідній системі може бути доведена формальними засобами), то тестувальник отримує в своє розпорядження досить потужний інструмент аналізу загальної цілісності системи. Працюючи з моделлю можна створити такі ситуації, які неможливо створити в тестовій лабораторії для реальної системи. На моделі програмного коду системи можна аналізувати його властивості й такі параметри системи, як оптимальність алгоритмів або її стійкість [10].

Як і інші методи тестування, перевірка на основі моделей має свої недоліки. У більшості випадків дуже складно автоматично побудувати модель на основі коду (є випадки, коли це можливо, але вони зустрічаються не часто). Створити модель вручну досить складно. Крім того, при вибухові розширення простору станів [8, 65] (умови, при яких з'являється дуже багато можливих станів кінцевого автомата) обсяг перевіряємої інформації збільшується до неприпустимого розміру. Розширення простору станів можна в деякій мірі контролювати, застосовуючи методи редукції зі складною евристичною логікою. Але й у них є свої мінуси: такі методи можуть призвести до того, що деякі дефекти будуть пропущені.

Нарешті, перевірка на основі моделей вимагає чіткого плану проектування та реалізації. Вона може підтвердити правильність проекту, але не гарантує відсутність помилок в реалізації. На практиці перевірка на основі моделей дієва лише для невеликих базових блоків програми. Крім згаданих методів, існують гібридні види тестування, що поєднують в собі динамічний аналіз і перевірку на основі моделей. Як приклад можна назвати інструмент CHESS, який буде розглянуто пізніше [23].

1.3 Моделювання і верифікація паралельних програм

1.3.1 Формальні методи і тестування програмного забезпечення

Розробка моделей та специфікацій пов'язана з «математизацією» програмування. Спроби використовувати різні математичні підходи для конструювання і навіть генерації програм розроблялися з перших років виникнення комп'ютерів. Відносний успіх був досягнутий в теорії компіляторів, реляційних баз даних і в декількох вузькоспеціальних областях; серйозних результатів в більшості практичних областей досягти не вдалося. До формальних методів у програмуванні почали ставитися скептично.

Новий сплеск інтересу до формальних методів стався в першій половині 90-х [18, 65]. Його викликали перші результати, отримані при використанні формальних моделей і формальних специфікацій у тестуванні.

Переваги тестування на основі моделей полягають в тому, що:

- тести на основі специфікації функціональних вимог більш ефективні, оскільки вони більшою мірою націлені на перевірку функціональності, ніж тести, побудовані тільки на знанні реалізації;
- на основі формальних специфікацій можна створювати тести, що перевіряються самі (self-checking), оскільки з формальних специфікацій часто можна витягти критерії перевірки результатів цільової системи.

Однак не було ясності відносно якості таких тестів. Моделі звичайно простіші в реалізації, тому можна було припустити, що тести, що добре «покривають» модель, занадто бідні для покриття реальних систем. Потрібні були широкі експерименти в реальних проектах.

Модель – деяке відображення структури й поведінки системи [8]. Модель може описуватися в термінах стану системи, входних впливів на неї, кінцевих станів, потоків даних та потоків управління, які повертаються системою результатів і т.д. Для відображення різних аспектів системи застосовуються і різні набори термінів. Формальна специфікація являє собою закінчений опис моделі системи і вимог до її поведінки в термінах того чи іншого формального методу. Для опису характеристик системи можна скористатися кількома моделями в рамках декількох формалізмів. Звичайно, чим більш загальною є нотація моделювання, тим більше труднощів виникає при автоматизації тестування програми на основі моделі/специфікації, описаної в цій нотації. Одні нотації і мови більш орієнтовані на доступність та прозорість опису, інші – на наступний аналіз і трансляцію, зокрема, трансляцію специфікації в тест. Робилися спроби розробки мови формальних специфікацій, який відповідає вимогам промислового використання (наприклад, методологія RAISE), проте широкого застосування вони не знайшли [26].

У широкому сенсі, під формальними методами розуміються будь-які спроби використання математичних підходів до розробки програмного забезпечення з

метою підвищення його якості. Як правило, для цього використовуються математичні моделі різних сутностей, що беруть участь у процесі розробки. Прикладами таких моделей можуть служити модель вихідних вимог до розроблюваної системи, модель архітектури системи або модель реалізації системи.

Одним з основних напрямків в області формальних методів є методи доказу коректності програм, такі як методи аналітичної верифікації та перевірки моделей. У цих методах доказ коректності програм будується за наступною схемою. Для даної програмної системи створюються математична модель вимог до системи і математична модель реалізації цієї системи. Після чого доводиться наявність відношення "задовольняє" між цими двома моделями, що і розглядається як доказ коректності програми (рис. 2). Існує також цілий набір варіацій підходу. Наприклад, в якості моделі вимог може виступати модель програмної системи більш високого рівня абстракції, або доказ коректності може зводитись до доведення несуперечності однією з математичних моделей.

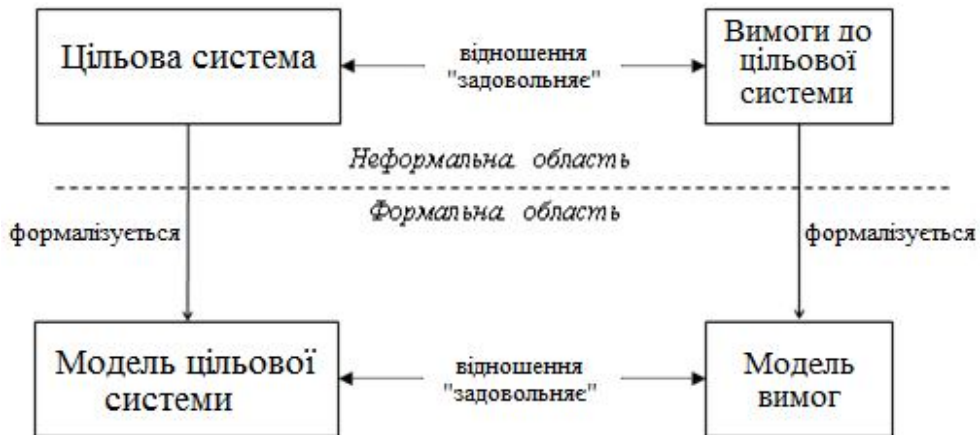


Рис. 2. Схема роботи методів аналітичної верифікації та перевірки моделей

Однак, незважаючи на великі зусилля, вкладені в розвиток даного напрямку, на даний момент так і не з'явилося методів доказу коректності програм, які змогли б знадати прийнятні рішення для забезпечення якості реальних програмних систем. У результаті, одним з основних засобів забезпечення якості програмних систем було і залишається тестування: при розробці програм з підвищеними вимогами до надійності частка витрат на тестування в бюджеті проекту може досягати 80%.

Численні дослідження в області формальних методів знайшли своє відображення і в розвитку нових технологій тестування. Один з напрямків, що активно розвивається, – тестування на основі моделей – вже встигло показати свої переваги в цілому ряді великих індустріальних проектів [65].

Тестування на основі моделей дозволяє систематизувати й автоматизувати процес розробки тестових наборів за допомогою використання різного роду математичних моделей. І в тих випадках, коли невеликого ручного тестування

виявляється недостатньо для забезпечення необхідного рівня якості програмного забезпечення, тестування на основі моделей дозволяє досягнути істотного підвищення якості тестування з витратами меншими, ніж при використанні інших підходів.

1.3.2 Процес верифікації моделей програм

Перевірка моделей (перевірка на моделі, англ. Model checking) – метод автоматичної формальної верифікації паралельних систем з кінцевим числом станів [8], дозволяє перевірити чи задовольняє задана модель системи формальним специфікаціям. Застосування методу перевірки на моделі складається з декількох етапів.

1. Моделювання. Перше завдання полягає у приведенні проектованої системи до такого формального вигляду, який був би прийнятний для інструментальних засобів верифікації моделей програм. У багатьох випадках це просто завдання компіляції. В інших випадках, при обмеженнях за часом та обсягом пам'яті, моделювання може зажадати абстракції, щоб позбутися від несуттєвих деталей, що не відносяться до справи.

2. Специфікація. Перед проведенням верифікації потрібно сформулювати властивості, якими повинна володіти проектована системи. Зазвичай специфікації задаються мовою формальної логіки. Для апаратури і програмного забезпечення, як правило, застосовують темпоральну логіку, що дозволяє описувати, як поведінка системи змінюється в часі.

Важливим питанням специфікації є повнота. Метод перевірки на моделі дає можливість переконатися, що модель проектованої системи відповідає заданій специфікації, проте визначити, чи охоплює задана специфікація всі властивості, якими повинна володіти система, неможливо.

3. Верифікація. В ідеальному випадку верифікація проводиться повністю автоматично. Однак на практиці вона часто вимагає сприяння людини. Однією зі сторін діяльності людини є аналіз результатів верифікації. Якщо результати перевірки негативні, то користувачеві нерідко надають трасу, яка містить помилку. Вона будується як контрприклад для перевірки властивості і може допомогти проектувальнику простежити, де виникає помилка. У цьому випадку аналіз помилкової траси може спричинити модифікацію системи і повторне застосування алгоритму перевірки на моделі.

Помилкова траса може з'явитися і в результаті неправильного моделювання системи, а також в результаті використання неправильної специфікації, у таких випадках говорять про помилкове спростування. Траса, що демонструє помилку, може виявитися досить корисною для виявлення цих двох проблем. Нарешті, не виключена можливість, що рішення задачі верифікації не буде завершено через великий розмір моделі, що не дозволяє розмістити її в пам'яті комп'ютера. У такому разі, можливо, буде необхідно провести верифікацію повторно, змінивши

параметри інструментального засобу, що реалізує перевірку на моделі, або спростивши саму модель (скажімо, за рахунок більш сильної абстракції) [8, 27].

1.3.3.2 Основні поняття мереж Петрі

Мережі Петрі представляють собою апарат моделювання динамічних дискретних систем і є одним з найбільш адекватних способів опису асинхронного виконання паралельних процесів, у тому числі в розподілених системах [11-13].

Слід зазначити, що, оскільки мережі Петрі призначені більше для моделювання систем, представлення на їх базі архітектури проектованої програмної системи, у тому числі паралельної, для деяких розробників виявляється дещо ускладненим. Це є наслідком не занадто високої узгодженості цієї моделі з популярними на даний момент методами програмування. Проте незважаючи на це, вони є могутнішим засобом побудови паралельних обчислень, що включають у себе можливості існуючих популярних засобів. Крім того, представлення обчислювального процесу у вигляді мережі Петрі може бути не прив'язане до самої реалізації її функціонування і може бути побудоване шляхом виділення підзадач в окремі функціональні блоки. На мережі Петрі в цій ситуації покладається лише задача організації послідовності їхнього виконання і синхронізація [28, 29].

1.3.3.1 Загальні відомості про мережі Петрі

Можна сказати, що мережа Петрі узагальнює поняття кінцевого автомата й додає йому деякі властивості, характерні мережам кінцевих автоматів. У якомусь роді мережа Петрі близька за змістом до діаграми станів автомата, що може перебувати одночасно в декількох станах. Кожний перехід зумовлюється якою-небудь підмножиною поточних станів і заміняє його іншою підмножиною станів. Позиції й переходи мережі Петрі в такій інтерпретації відіграють роль станів автомата та дій на переходах відповідно.

Мережею Петрі [11] називається п'ятірка $N = (P, T, F, H, M_0)$, в якій P – деяка множина вершин місць, T – множина переходів, F та H – функції інцидентності, $F : P \times T \rightarrow \{0,1\}$, $H : T \times P \rightarrow \{0,1\}$. Неформально функцію F можна трактувати як множину направлених дуг, які ведуть із місць до переходів, а H – множину направлених дуг, які ведуть від переходів до місць. Компонента M_0 – в означенні початкова розмітка, яка полягає в співставленні кожному місцю деякої кількості так званих фішок.

Графічно мережа Петрі представляється у вигляді дводольного орієнтованого графа із двома типами вершин – позиціями (вершина місць) й переходами. Позиції позначаються колами, переходи – перпендикулярною шляху рисою або прямокутником. Як правило, рисою позначають простий, миттєвий перехід, прямокутником – тривалий. У позиціях можуть перебувати фішки – сутність, наявність якої говорить про те, що умова, що відповідає поточній позиції,

виконується. Приміром, наявність фішки може говорити про наявність вхідних даних для деякої процедури. Наявність декількох фішок говорить про те, що умова виконана з багаторазовим запасом. У прикладі із процедурою це може бути наявність декількох елементів у черзі її вхідних даних. Кількість фішок у кожній позиції є цілим ненегативним числом. Наявність фішок у позиції на графі позначається числом або жирними крапками у відповідній кількості. Сукупність всіх фішок, розміщених у позиціях мережі Петрі, називається розміткою мережі.

Дуги мережі Петрі можуть бути спрямовані лише від позицій до переходів або від переходів до позицій. Дуги можуть бути кратними, тобто мати вагу – також ціле ненегативне число. Наявність кратної дуги між позицією й переходом еквівалентно наявності між ними відповідної кількості простих дуг у тому ж напрямку. Дуги, спрямовані від позицій до переходів, називаються вхідними, спрямовані від переходів до позицій – вихідними.

Аналогічними термінами позначаються й відповідні позиції стосовно деякого переходу.

Таким чином, мережа Петрі повністю описується наступними параметрами:

- множина позицій;
- множина переходів;
- множина вхідних дуг (дуг від позицій до переходів);
- множина вихідних дуг (дуг від переходів до позицій);
- множина фішок, розміщених у позиціях з початку (початкова розмітка) [11, 13].

У мережах Петрі вершина переходу асоціюється з виконанням певної дії, а час знаходження мітки у вершині відповідає часу виконання відповідної дії. Поява мітки у вершині місця відповідає закінченню виконання дії, що відображується вхідною вершиною переходу, і говорить про можливість спрацьовування вихідної вершини переходу. Мітка у вершині місця також може відповідати виконанню певної попередньої умови і певному виконанню дії, що пов'язана з даною умовою.

Розташування міток у вершинах місць мережі Петрі визначає її поточний стан, який може бути сталим і несталим [11].

Сталим називають стан мережі, в якому своє відпрацювання закінчили всі вершини переходів і мітки з вхідних по відношенню до кожного переходу вершин місць перейшли у вихідні.

Несталим називають стан мережі, в якому є вершини переходів, які не завершили свого відпрацювання. У цьому випадку виникає ситуація видалення мітки з вхідної вершини місця і затримки її передачі у вихідну вершину.

У логічних схемах нерідко використовуються умови заборони. У графах Петрі такі елементи реалізуються за допомогою інверсних дуг чи дуг заборони. Вони зображуються дугою, що закінчується великою крапкою, як показано на рис. 3.

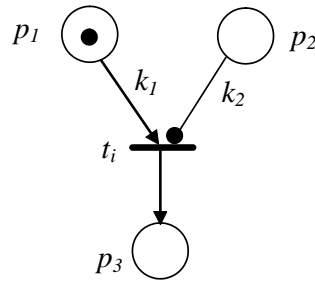


Рис. 3 Позитивні (k_1) та інверсні (k_2) дуги у мережах Петрі

Дуги заборони k_2 можуть бути спрямовані тільки від вершин місць до вершин переходів, що пояснюється можливістю появи заборони тільки після умови, а не після дії. Дуги, що закінчуються стрілкою на кінці, як наприклад k_1 , є позитивними. Наявність дуг заборони у мережі Петрі викликає появу додаткових правил спрацювання переходів t_i .

Правило спрацювання переходів з інверсними дугами: перехід t_i з інверсною дугою буде активізованим тільки тоді, коли у кожній вхідній вершині місця з позитивною дугою є мітка, а у кожній вхідній вершині місця із дугою заборони мітка відсутня.

Приклад активізованого переходу показаний на рис.3, де у вхідній вершині місця p_1 є мітка, а у вхідній вершині місця p_2 з інверсною дугою мітка відсутня, після спрацювання переходу мітка перейде у вихідну вершину місця p_3 .

1.3.3.2 Види мереж.

Класичні мережі Петрі мають різні інтерпретації в залежності від області застосування, яка визначає набір понять і правил їх побудови. Широке розповсюдження мереж Петрі призвело до їх використання дослідниками для розв'язання задач у багатьох областях. Так були сформовані Е-мережі (EN), Макро-Е-мережі (MEN), О-мережі (O-Net), Про-мережі (Pro-Net), кольорові мережі (CPN), часові мережі (TP), біполярні мережі (BP), автоматні мережі (SM), МG-мережі, вільно-вибрані мережі (FC), сильнозв'язані мережі (SC), оціночні мережі (BPN), управляючі мережі (SN), регулярні та багато інших мереж [11].

На сьогоднішній день сформовані загальні структурні і функціональні елементи відомих інтерпретацій мереж Петрі (PN), чітко визначені правила побудови і відпрацювання цих інтерпретацій [29].

Інтерпретацією мережі Петрі називається визначений набір понять і правил, в основі яких лежить принцип побудови розмічених графів, що складаються з двох типів вершин – вершин місць $T = \{t_i\}$ та вершин переходів $P = \{p_\epsilon\}$ – з'єднаних між собою направленими дугами $K = \{k_q\}$ за певними функціональними правилами S [11].

Поява нових інтерпретацій та модифікацій мереж Петрі пов'язана з намаганням позбавитися від їх критичних якостей та із широким застосування мереж Петрі для відображення об'єктів у багатьох областях науки і техніки.

Відомі інтерпретації мереж Петрі для алгоритмічного опису та моделювання паралельних процесів у залежності від особливостей їх алгоритмічної побудови поділяються на три групи: числові (EN) та макрочислові (MEN) мережі; оціночні (обмежені) мережі – BPH; безпечні мережі (SPN) [11].

Особливістю числових мереж Петрі є розподіл вершин переходів $T = \{t_i\}$ на декілька типів з різноманітною структурою, а в макрочислових мережах Петрі для кожного типу вершин структура вибирається за допомогою спеціального макросу. В таких мережах Петрі з'являється нестандартна вершина у вигляді шестикутника. Відмінністю цього типу мереж є модульний принцип, який лежить в основі їх побудови, спрощує і полегшує процес створення моделі на базі EN та MEN-мереж. Недоліком цих інтерпретацій мереж Петрі є необхідність побудови нового модуля при навіть невеликій зміні структури існуючого модуля. Це приводить до значних апаратних і програмних витрат для їх реалізації.

До оціночних належать інтерпретації мереж Петрі, в яких вершини місць $P = \{p_\varepsilon\}$ вміщують ціле число міток N ($N \geq 1$), а оціночні дуги мають вагу, яка показує, скільки міток необхідно накопичити у вершині місця p_ε , щоб спрацювала наступна вершина переходу t_i . Для спрацьовування переходу встановлене наступне правило.

До безпечних мереж Петрі (SPN) належать інтерпретації, у яких вершини місць $P = \{p_\varepsilon\}$ при спрацьовуванні переходів t_i не можуть накопичувати більше однієї мітки. Ці мережі найбільш розповсюджені при розв'язанні багатьох технічних задач, оскільки найбільш просто відображають структуру і функціонування об'єктів, що за допомогою них моделюються.

Безпечними мережами Петрі (SPN) є фактично класичні мережі Петрі, в яких вершини місць $P = \{p_\varepsilon\}$ можуть мати одну мітку або бути немаркованими, а дорівнює одиниці, а вага дуг дорівнює одиниці.

Ці мережі отримали широке розповсюдження завдяки простоті і наочності подання структури та відображення функціонування систем управління з паралелізмом, а також зручності формалізації правил побудови мереж та простоті автоматизації процесу побудови, перетворення та відлагодження моделей. Для реалізації вони потребують незначні програмні та апаратні ресурси, що дозволяє використовувати безпечні мережі Петрі для моделювання об'єктів в реальному часі. Цей клас мереж Петрі має найбільше число інтерпретацій. Особливо ефективним є застосування безпечних мереж Петрі для побудови та аналізу функціонування алгоритмів складних керуючих систем.

1.3.3.3 Принцип функціонування мереж Петрі

Функціонування мережі Петрі здійснюється у вигляді послідовності спрацьовування переходів. У кожний момент часу в мережі є деяка кількість дозволених переходів, тобто таких, які можуть спрацювати. Перехід вважається дозволеним, якщо у всіх його вхідних позиціях кількість фішок не менше, ніж

кратність відповідних вхідних дуг. Із безлічі дозволених переходів спрацювати може кожен. Який саме спрацює, визначається поза мережею, також як і момент його спрацювання в часі. Вибір може бути здійснений на основі очікування апаратної події, події завершення тривалого процесу, вибору користувачем і т.п. При спрацюванні простого переходу з кожної його вхідної позиції вилучається кількість фішок, рівна кратності відповідної вхідної дуги, після чого до кожної вихідної позиції додаються фішки відповідно до кратності вихідних дуг.

Після кожного спрацювання переходу множина дозволених переходів у загальному випадку міняється, оскільки міняється поточна розмітка мережі. Вважається, що мережа Петрі "жива", якщо в ній є дозволени переходи. Якщо після спрацювання чергового переходу дозволених переходів у мережі не залишається, вона завершує виконання.

У випадку, коли позиція є вхідною для двох або більше дозволених переходів, спрацювання кожного з них і відповідне зменшення кількості фішок у ній може заборонити інші переходи. Такою ситуацією моделюються конфлікти, коли для виконання декількох операцій потрібне використання загального ресурсу. Приклад такої ситуації видний на рис.4. На самому початку роботи мережі дозволеним є один перехід – t_1 . Після його першого спрацювання дозволеними є наступні два переходи мережі - t_2 і t_3 . Якщо ж тепер спрацює, приміром, перехід t_2 , він заборонить перехід t_3 , і навпаки, спрацювання t_3 заборонить перехід t_2 , оскільки переходи t_2 і t_3 мають загальну вхідну позицію p_3 .

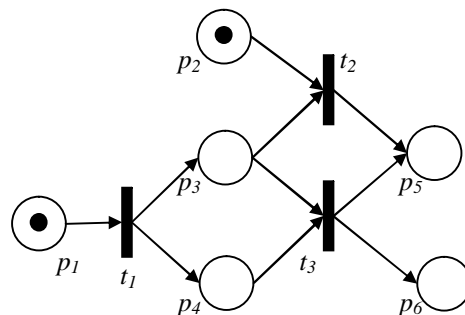


Рис. 4. Приклад мережі Петрі.

Спрацювання простого переходу вважається миттєвим. Оскільки ймовірність одночасного походження двох миттєвих подій дорівнює нулю, два простих переходи не можуть спрацювати одночасно. Звідси випливає досить парадоксальна властивість мереж Петрі: при тому, що вони моделюють асинхронне виконання паралельних процесів, сама по собі робота мережі Петрі відбувається строго послідовно. Саме неможливістю одночасного спрацювання будь-яких переходів визначається асинхронна природа мереж Петрі. Паралелізм же виконання виражається в тім, що в один момент часу різні ділянки мережі можуть відтворювати виконання різних незалежних процесів.

1.3.3.4 Аналіз мереж Петрі

Статичні характеристики стосуються топології побудови мережі Петрі і залежать від її модифікації чи інтерпретації [11]. При розгляді мереж Петрі, на основі яких моделюються паралельні алгоритми, виділимо наступні статичні характеристики:

- дуги мережі Петрі можуть з'єднувати лише різнойменні вершини, тобто вершину переходу з вершиною місця, чи навпаки;
- мережа Петрі буде замкненою, якщо через кінцеве число спрацьовування переходів вона відновлює початкову розмітку;
- ділянку мережі Петрі, яка відображає низку послідовних дій, що активізуються однією міткою можна замінити одним макропереходом.

Останню характеристику можемо пояснити так: у мережі Петрі, яка моделює послідовність дій (послідовний алгоритм) відсутні критичні властивості, тому на етапі дослідження властивостей функціонування деталізувати такі ділянки не потрібно.

До динамічних характеристик мереж Петрі належать властивості живості, безпечності та неконфліктності [11]. Розглянемо ці властивості для безпечних мереж Петрі, які можуть вміщувати в окремій вершині місця не більше 1-ї мітки.

Властивість *живості* проявляється в мережі Петрі, коли при заданому максимальному розміточному класі \bar{M}_{\max} для кожної розмітки $\bar{m}_i \in \bar{M}_{\max}$ існує така послідовність спрацьовування переходів δ_γ , в результаті якої кожна вершина переходу t_i може спрацювати. Приклад неживої мережі Петрі представлений на рис. 5, а, де при спрацьовуванні переходів $\delta = t_1, t_2, t_3$ чи $\delta = t_1, t_2, t_4$ початкова розмітка $\bar{m}_0^N = (1,1,0,0)$ у замкненій мережі Петрі не може з'явитися знову. Тому вершина переходу t_1 не може бути активована вдруге. Причиною виникнення властивості неживості є початок розгалуження паралельних ділянок від вершини місця, яка не множить мітку в розгалуженні, а лише спрямовує її в одну з паралельних гілок, тому в результаті спрацьовування переходів лише однієї гілки мітка може з'явитися у вершині місця p_1 або p_2 . На рис. 5, б та 5, в попередня мережа перетворена в живу, що забезпечується введенням розгалуження від вершин переходу t_2 (б) чи макропереходу τ_2 (в), які забезпечують розмноження міток і активують низки переходів у паралельних гілках. В результаті спрацьовування гілок, мітки можуть з'явитися в обох вершинах місць p_1 та p_2 , що приведе до запуску вершини переходу t_1 . Властивість неживості в алгоритмічних конструкціях проявляється як неможливість спрацьовування певних переходів чи ділянок мережі. Якщо така мережа виконує свої функції, то неживі переходи є зайвими, їх потрібно видалити. У випадку невиконання певних функцій по причині неспрацьовування певних переходів чи ділянок мережі потрібно перебудувати структуру мережі, забезпечуючи живість всіх представлених в алгоритмі конструкцій.

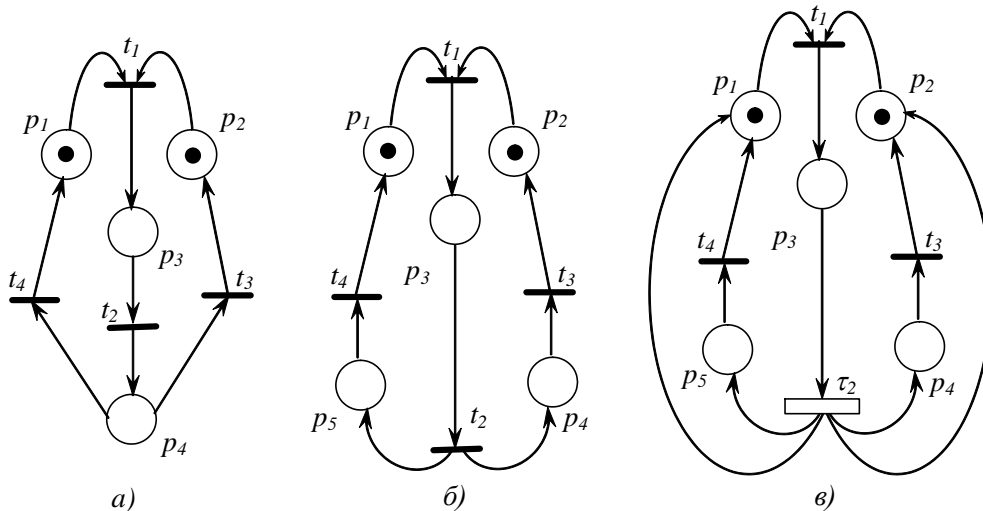


Рис. 5. Приклад неживої (а) та живої (б, в) управляючих мереж Петрі.

Властивість *безпечності* управляючої мережі Петрі маємо у тому випадку, коли розмітка $m(p_\varepsilon)$ вершини місця мережі Петрі буде безпечною, тобто при будь якій послідовності спрацьовування переходів δ_γ значення $m(p_\varepsilon) \leq 1$ у кожній вершині місця. Це значить, що в управляючих мережах Петрі попадання двох міток в одну вершину місця $p_\varepsilon \in$ забороненим.

Небезпечною мережа Петрі (рис. 6, а) стає по причині можливості попадання у вершину місця p_3 двох міток одночасно. Для усунення небезпечної якості введемо замість переходу t_1 керований макроперехід τ_1 , який не дозволить переміщувати мітку у p_3 коли вона розмічена.

Конфліктні ситуації в управляючій мережі Петрі можуть виникнути у випадку, коли з вершини місця p_ε виходять декілька дуг (рис. б) і в ній не визначена вхідна керуюча вектор-функція [11].

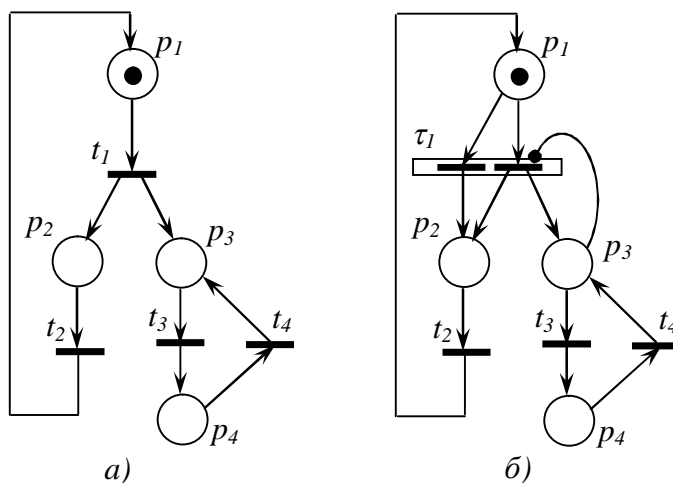


Рис. 6. Приклад небезпечної (а) та безпечної (б) мережі Петрі.

В такому випадку немає чіткого алгоритмічного шляху подальшого руху мітки і мережа може „зависнути”. Розмітка мережі називається безконфліктною, якщо при всіх розмітках $\vec{m}^{(n)} \in \vec{M}_{\max}$ не настає конфліктних ситуацій.

Конфліктна ситуація показана на рис. 7, а, де можуть бути активізовані переходи t_1 або t_2 , але спрацьовування одного з переходів приводить до неможливості спрацьовування переходу t_3 . Це приводить до ситуації „зависання” мережі. Наприклад, при виборі вершин переходів в порядку їх зростання, активованим буде перехід t_1 , мітка з’явиться у вершині p_2 , але ніколи не з’явиться у вершині місця p_3 , що приводить до неможливості спрацьовування вершини переходу t_3 . Таку ситуацію можемо розв’язати введенням керування у вершині місця p_1 та керованого макропереходу τ_3 , що складається з двох простих переходів t_3 та t_4 , який забезпечить вибір подальшого шляху мітки через вершину переходу t_1 чи t_2 , а також виконання роботи, передбаченої макропереходом τ_3 (рис. 7, б).

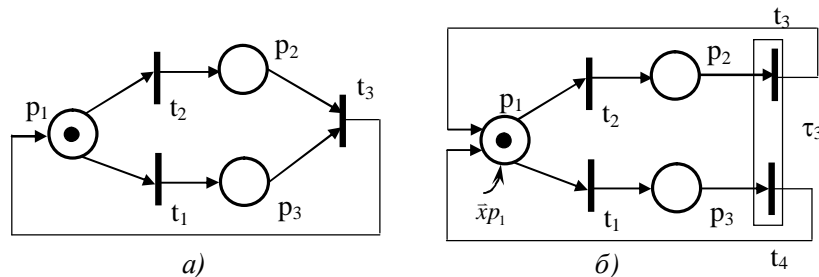


Рис. 7. Конфліктна ситуація (а) в мережі Петрі та один з варіантів (б) її усунення.

Мережа Петрі називається некритичною тільки тоді, коли вона не має неживих, небезпечних та конфліктних розміток [11]. Наявність критичних якостей не дозволяє якісно проводити машинні експерименти з нею. Критичні ситуації можуть вказувати на протиріччя в структурі чи логіці побудови моделі. Для автоматичного аналізу та усунення критичних якостей потрібно локалізувати критичну властивість, тобто обмежити ділянку, яка буде замінена іншою алгоритмічною конструкцією, та підібрати алгоритмічну конструкцію, яка її замінить без порушення логіки функціонування мережі. Для реалізації цих задач розглянемо спочатку представлення основних алгоритмічних конструкцій за допомогою мереж Петрі [29].

1.3.4 Автоматична побудова моделей в термінах мереж Петрі

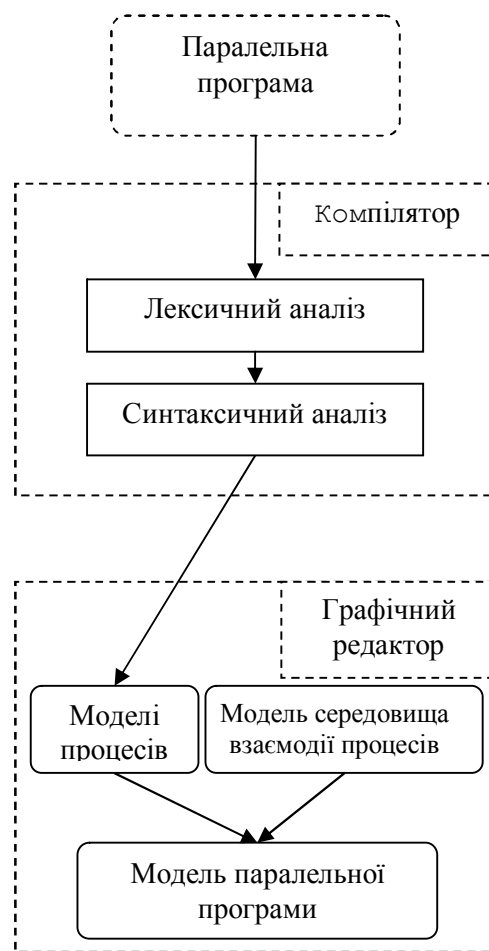
Для специфікації паралельних програм, запрограмованих імперативними мовами програмування, досить описати потік керування й потік даних. Мережі Петрі вважаються точним і наочним формалізмом для подання потоку керування в паралельній програмі. Однак при побудові моделей у термінах мереж Петрі

виникають труднощі при відображенні стандартних для більшості мов програмування даних і при описі їхньої модифікації.

Підходи до роботи з даними в моделях, що представляються мережами Петрі, можна класифікувати в такий спосіб. По-перше, дані і їхня модифікацію можна залишати за межами мереж Петрі й відображати тільки потік керування паралельної програми. Такий підхід реалізують ряд графічних систем моделювання й аналізу розподілених систем [9, 30]. По-друге, можна використовувати спеціалізовану мову опису даних (наприклад, мова у середовищі CPN-Design). Застосування цього підходу вимагає вивчення питання про ідентичність використання даних і операцій над ними в принципово різних мовах. По-третє, дані і їхня модифікація можна залишати в термінах вихідної мови програмування. Такий підхід вимагає адаптації мереж Петрі для опису структури паралельних програм з використанням типів даних з більшою кількістю значень, описаних у термінах імперативних мов програмування.

Пропонований метод розроблений у рамках третього підходу й ґрунтується на поданні моделі в термінах мереж Петрі для програмування. Відповідно до обраного підходу метод побудови моделі паралельної програми складається з наступної послідовності дій: потік керування паралельною програмою перетворюється в структуру мереж Петрі; процеси паралельної програми разом з даними перетворюються в маркери мереж Петрі; потік керування отриманої моделі контролюється даними, опис операцій над якими залишається в термінах вихідної мови програмування.

Рис. 8. Фази побудови моделі паралельної програми



Автоматична побудова моделі паралельної програми пропонується виконувати за схемою, представленою на рис.8. На ньому показано, що кінцева модель паралельної програми представляється композицією моделей виконання окремих послідовних процесів програми й моделі міжпроцесорної взаємодії. Така розбивка вихідної програми на частини обумовлена тим, що розглянутий клас паралельних програм являє собою набір послідовних взаємодіючих між собою

процесів, у якому процеси визначають загальний алгоритм обчислень, а функції організації взаємодії реалізують тільки передачу даних між процесами й не можуть бути інтегровані в алгоритм розв'язуваного завдання.

Побудова моделей окремих послідовних процесів програми укладається в схему роботи класичного компілятора. Тут можна виділити фази розбору текстів паралельної програми з метою виявлення окремих послідовних процесів і генерації деякого проміжного подання. Крім того, при побудові моделей окремих процесів можливе виконання фази часткової редукції вихідної програми залежно від цілей моделі (критерій редукції задається як вхідний параметр компілятора). Так, наприклад, у випадку аналізу моделі на виявлення помилок міжпроцесорної взаємодії функції, що не викликають прямо або побічно функції з бібліотеки середовища організації взаємодії, можна розглядати як неподільну подію. На етапі генерації коду виробляється співставлення семантичним конструкціям окремих процесів паралельної програми відповідних шаблонних конструкцій у термінах мереж Петрі для програмування.

Розглянутий метод припускає наявність графічного редактора мереж Петрі для відображення побудованих моделей. Крім того, передбачається, що редактор повинен надавати можливості неавтоматичної побудови й зміни моделей у термінах мереж Петрі [9].

1.4. Огляд засобів моделювання паралельних програм

1.4.1 Тестування на основі моделі

В наш час існує велика різноманітність абстрактних моделей програм, формальних мов специфікацій і алгоритмів перевірки специфікацій на моделях. Тут, швидше, загрожує *embarras de richesse*: у світі вже є понад двадцять діючих систем верифікації програм, побудованих за принципом *model checking*, а також багато десятків самих різних алгоритмів перевірки виконуваності специфікацій на моделях програм [8].

Розглянемо докладніше інструменти, що здійснюють верифікацію програм на основі моделі:

- BLAST – статичний аналізатор C програм [31], [32];
- CADP (Construction and Analysis of Distributed Processes) – інструмент проектування протоколів та розподілених систем [33];
- CHESSE – інструмент для тестування багатопоточних .Net (керованих) і Win32, 64 програм [34], [35], [23];
- ISP - верифікатор коду MPI програм [36];
- Java Pathfinder – вільний інструмент для перевірки багатопоточних Java програм [31], [37], [38];
- MoonWalker – вільний інструмент для перевірки .Net програм [39];
- MRMC (Markov Reward Model Checker) [40], [41];
- NuSMV - символічний модел-чекер [42], [43];
- PRISM - ймовірнісний, символічний модел-чекер [44], [45];
- Rabbit – модел-чекер для систем реального часу [46], [47];
- SPIN – модел-чекер загального призначення для верифікації коректності розподілених програм [48], [49];
- Vereofy – модел-чекер програм компонентних систем [50], [51];
- ?CRL2 – вільний інструмент, заснований на ACP [52];
- UPPAAL – інструментарій для моделювання, верифікації та валідації систем реального часу модельованих як мережі тимчасових автоматів [53], [54];
- та інші [27], [55].

Отже, розглянемо деякі з перерахованих засобів детальніше.

Berkeley Lazy Abstraction Software Verification Tool (BLAST) [31], [32] – програма перевірки моделей для мови C. Завдання, яке вирішується інструментом BLAST – це перевірка того, що програма задовольняє поведінковим вимогам до неї. BLAST реалізує підхід абстракції і уточнення по контрприкладом (англ. *counterexample-driven automatic abstraction refinement*) для конструювання абстрактної моделі, яка потім перевіряється на властивості безпеки (англ. *safety*). Абстракція будується по ходу аналізу і тільки до необхідної точності, яка встановлюється в ході аналізу.

CHESS [23], [34], [35] – засіб, який розроблений фахівцями Microsoft, поєднує в собі перевірку на основі моделей і динамічний аналіз. Цей інструмент виявляє помилки розпаралелювання шляхом систематичного аналізу планування і чергування потоків. Він здатний виявляти проблеми, пов'язані з перегонами, взаємоблокування, зависаннями, активними блокуваннями і пошкодженням даних. Також спрощує усунення неполадок, оскільки дозволяє повністю відтворювати результати. Як і у випадку будь-якої іншої перевірки на основі моделей, систематичні дослідження забезпечують повне охоплення коду тестами.

CHESS, будучи засобом динамічного аналізу, виконує звичайний модульний тест циклічно за допомогою спеціалізованого планувальника. При кожному повторі обирається новий порядок планування (розподілу процесорного часу між потоками). З іншого боку, це засіб перевірки на основі моделей, і він управляє спеціалізованим планувальником, здатним створювати спеціальні чергування потоків. Для контролю розширення простору станів у CHESS застосовується часткова редукція і обмеження кількості нових контекстів ітерацій.

Використовуючи обмеження числа контекстів ітерацій, CHESS не зменшує глибину розширення простору станів, а скорочує кількість перемикачів потоків в даному сеансі роботи програми. Сам потік може виконувати будь-яку кількість дій між перемикачними, причому глибина виконання залишається необмеженою (це велике досягнення у порівнянні з традиційною перевіркою на основі моделей). Такий принцип роботи заснований на емпіричних даних про те, що для виявлення більшості помилок паралельної обробки достатньо невеликого числа систематичних перемикачів потоків.

CHESS дозволяє виявляти взаємоблокування і гонки (застосовується алгоритм Goldilocks), але результати залежать від того, наскільки якісно програміст перевіряє інші стани. Крім того, спочатку передбачається, що всі програми повинні завершитися і що всі потоки більш-менш рівноправні. Тобто, якщо програма входить у нескінченний цикл, інструмент повідомляє про активне блокування.

Тестування починається з того, що CHESS запускається при межі числа контекстів ітерацій, що дорівнює 2. Як тільки будуть виявлені всі можливі помилки, межа розширюється до 3 і т. д. Досвід показує, що більшість помилок виявляється при значеннях 2 і 3. Отже, таким чином можна прискорити пошук помилок.

Що стосується оснащення засобами моніторингу та протоколювання (instrumentation), то CHESS обходить виклики синхронізуючих Win32-функцій і навмисно вводить недетермінованість. Крім того, цей засіб вимагає, щоб в коді було досить багато операторів assert, що гарантують узгодженість станів (у грамотно написаному коді так і має бути). Однак, як і більшість засобів динамічного аналізу, CHESS забезпечує гарне охоплення тільки при наявності якісного комплексу тестів. CHESS – віддушина для тих розробників і тестуваль-

ників, які покладаються при тестуванні чергувань тільки на стресові тести. Це засіб виконує регулярні тести модулів і методично моделює потрібні чергування.

Java PathFinder (JPF) [31], [37], [38] – вільний інструмент для перевірки багатопоточних Java-програм. За своєю суттю це віртуальна Java-машина (англ. Java Virtual Machine) на основі якої реалізовано методи перевірки моделей (англ. model checking). Це означає, що JPF виконує програму не один раз, як це робить звичайна віртуальна машина, а по всіх можливих шляхах, пов'язаних з перемиканням потоків планувальником. JPF знаходить такі помилки як тупики, необроблені винятки, а також порушення умов, що задаються користувачем у вигляді `assert` виразів. Крім того, користувач може писати спеціальні «слухачі» (англ. listener-extensions) для перевірки довільних властивостей. Деякі з таких «слухачів», такі як стану гонок і обмеження на купу (англ. heap bounds) поставляються разом з JPF. При знаходженні помилки JPF виводить повну трасу, яка до неї веде, включаючи всі необхідні перемикання планувальника.

У загальному випадку JPF здатний перевіряти будь-які Java програми, які не залежать від не підтримуваних `native`-методів. Віртуальна машина JPF не може виконувати платформенно залежний код. Це накладає суттєві обмеження на те, які стандартні бібліотеки можуть використовуватися додатком, що тестується. Хоча в принципі можливо написати для цих бібліотек спеціальні обгортки (використовуючи `Model Java Interface`), але на даний момент в JPF немає підтримки `java.awt`, `java.net`, і лише обмежена підтримка `java.io` і `reflection`. Інше обмеження JPF – за місцем, що необхідне для зберігання стану, що обмежує розміри перевірюваних додатків до ~ 10 кб (залежно від їх внутрішньої структури). Для вирішення проблем масштабування JPF надає гнучкі механізми розширення, які дозволяють підлаштовувати його під конкретні програми, але перевіряються властивості. Крім того, дані механізми дозволяють розглядати JPF як `framework` для різного роду технік верифікації. Через обмеження на бібліотеки та розмір додатків JPF до теперішнього часу використовувався для додатків, які є моделями, але написані на повноцінній мові програмування Java.

JPF розроблений в NASA. Розповсюджується під вільною ліцензією NASA Open Source Agreement version 1.3.

KISS [23] – інструмент для перевірки на основі моделей, розроблений в Microsoft Research для паралельних програм на C. Оскільки в системах паралельної обробки розширення простору станів відбувається дуже швидко, KISS перетворює паралельний додаток на C в послідовний і моделює чергування. Після цього виконується власне аналіз із застосуванням послідовного засобу перевірки на основі моделей.

Додаток оснащується виразами, що дозволяють перетворити паралельну програму в послідовну, причому KISS бере на себе контроль недетермінованості. Недетерміноване перемикання контекстів відбувається за тим же принципом, що і в CHESS, про який говорилося вище. Програміст повинен включити в код затвердження (`asserts`) перевіряючі вихідні припущення про паралельність

виконання. Цей засіб не дає помилкових позитивних результатів. Як прототип, застосовувався групою розробників драйверів для Windows, в яких в основному використовується C.

Zing [23],[56] – засіб перевірки на основі моделей, призначений для тестування проектів паралельних додатків. У Zing є власна мова для опису складних станів і переходів. Це засіб має все необхідне для моделювання паралельних кінцевих автоматів. Як і інші засоби перевірки на основі моделей, Zing – повноцінне рішення для перевірки проектів програм; він також допомагає переконатися в якості проекту, оскільки забезпечує перевірку вихідних припущень і формальний доказ наявності або відсутності тих або інших умов. Це засіб також дозволяє контролювати розширення простору станів, застосовуючи сучасні методики редукції.

Модель, яка використовується в Zing для перевірки коректності програм, створюється або трансляторами, або вручну. Транслятори для специфічних предметних областей написати можна, а от повноцінного та ефективного транслятора для некерованих або керованих додатків поки немає. З цієї причини Zing навряд чи стане в нагоді у великих проектах – хіба що для перевірки правильності окремих частин.

Spin [48],[49],[57]. Spin використовувався для верифікації деяких алгоритмів синхронізації для ядра Linux і для Threading будівельних блоків. Spin – це, напевно, самий старий і ґрунтовний інструмент такого роду. Його коріння сягає початку 80-их років, по ньому написано ряд книг, і, він активно розвивається і до сьогоднішнього дня. Spin включає безліч варіантів перевірки – динамічна перевірка із запам'ятовуванням станів, без запам'ятовування, повна перевірка моделі програми, не повна перевірка моделі програми (для дуже великих програм) і т.д.. Компілятор Promela (мова, яка використовується Spin) і верифікатор (Protocol ANalyser, PAN у термінології Spin) мають безліч ключів, керуючих різними аспектами роботи (режим перевірки, ступінь деталізації виведення, ліміт пам'яті і т.і.), так само є кілька GUI оболонок.

Сам процес роботи із Spin полягає в тому, що описується тест на спеціальній мові Promela (мова а PROcess MEta LAnguage), далі він компілюється і на виході отримуємо вихідний файл на мові C, який треба скомпілювати C компілятором, що б отримати верифікатор. Далі необхідно запустити верифікатор, і при виявленні помилки він створює файл з докладним описом помилки та історії виконання. Потім з цього файлу можна згенерувати Postscript файл для перегляду, або використовувати для "програвання" історії виконання.

Тим не менше Spin має деякі недоліки. По-перше, це спеціальна мова Promela для опису тестів; так як при переписуванні програми вручну на іншу мову, є ризик протестувати не зовсім те, що хотілося. По-друге, це послідовно узгоджена (sequentially consistent) модель пам'яті – підтримка вільного доступу до пам'яті ("розслабленої моделі пам'яті") просто необхідна для верифікатора алгоритмів синхронізації. По-третє, це відсутність вбудованої підтримки для таких

специфічних речей як виклики Win32 API WaitForMultipleObjects () або SignalObjectAndWait (), або помилкові пробудження на змінній умови POSIX, або очікування з таймаут, і так далі.

З вітчизняних розробок можна згадати сімейство інструментів розробки тестів на основі моделей з використанням кінцевих автоматів – UniTesK, що пропонується Інститутом системного програмування РАН в кооперації з компанією AST.

UniTesK (Unified Testing & specification toolKit) [58], [59] – технологія тестування програмного і апаратного забезпечення на основі формальних специфікацій, розроблена в Інституті системного програмування РАН. Технологія являє собою поєднання добре зарекомендованих технік, які можуть застосовуватися в різних комбінаціях, взаємно поєднуючись і підсилюючи одна одну. Це робить технологію гнучкою і налаштовуваною під існуючі процеси розробки на всіх етапах життєвого циклу розробки програмного забезпечення від збору та аналізу вимог до супроводу.

Основою для винесення вердиктів про правильність поведінки тестованої системи становлять контрактні специфікації у формі перед- і постумов, написані на розширених традиційних мовах програмування, таких як C, Java, і які дозволяють виносити вердикт повністю автоматично. Специфікації являються поданням функціональних вимог до системи. Форма специфікацій і засновані на них критерії покриття забезпечують простежуваність вимог.

Успішно використовуються на практиці, техніка побудови тестів на основі обходу графів станів, які дозволяють істотно мінімізувати кількість створюваного вручну програмного коду, разом з тим забезпечуючи різноманітність і масивність тестового набору.

Техніки абстракції даних і критерії покриття, засновані на вимогах, дозволяють гнучко керувати розміром тестового набору і направляти генерацію на покриття певних вимог, мінімізуючи тим самим час виконання тестового набору.

Спеціальний проміжний рівень, який є в технології, дозволяє швидко настроювати тестовий набір на різні реалізації з тією ж функціональністю.

Всі ці техніки забезпечують високу якість тестування, простежуваність вимог і високий рівень повторного використання компонентів тестового набору при мінімумі ручної роботи і прийнятному часу виконання тестів.

1.4.2 Інші засоби тестування паралельних програм

Розглянемо інші засоби для тестування паралельного виконання [23], [31], що допомагають усувати проблеми, пов'язані з взаємоблокуванням, голодуванням та іншими проблемами, що виникають при виконанні паралельних операцій. Кожен з засобів, які тут розглядаються, корисний у своїй галузі.

Intel Thread Checker [23], [60] – засіб динамічного аналізу, що дозволяє виявляти взаємоблокування (у тому числі потенційні), зависання, гонки і випадки

некоректного використання Windows-функцій синхронізації. Thread Checker вимагає, щоб вихідний код або скомпільований двійковий файл був оснащений засобами моніторингу та протоколювання, які дозволили б спостерігати кожне посилання на область пам'яті і кожен синхронізуючий Win32-примітив. Під час виконання такий двійковий файл надає аналізаторові достатньо інформації для побудови часткового порядку виконання операцій (partial-order of execution). Після цього виконується аналіз за алгоритмом «сталосся до» («happens-before»).

Для більшої продуктивності і масштабованості цей інструмент зберігає інформацію лише про самі останні звернення до загальної змінної. Це допомагає підвищити його ефективність при аналізі додатків, що виконують тривалі операції. Побічний ефект такого підходу – пропуск частини помилок. Втім, можливо, важливіше виявити більшість помилок у довгостроково виконуваних додатках, ніж всі помилки в додатках, що запускаються лише на дуже короткий проміжок часу.

Єдиний великий недолік даного інструменту в тому, що він не може враховувати синхронізацію через interlocked-операції, які використовуються, наприклад, в спін-блокуванні. Втім, серед засобів тестування паралельних програм, що застосовують тільки стандартні синхронізуючі примітиви, цей інструмент, мабуть, підтримується найкраще.

RacerX [23] – засіб статичного аналізу для виявлення гонок і взаємоблокувань. Дозволяє обійтися без скрупульозного коментування всього вихідного коду. Єдина потрібна інформація – таблиця, в якій вказуються API-функції, що використовуються для отримання і звільнення блокувань. Там же можна вказати атрибути примітивів блокування, такі як кручення, блокування, реєнтерабельність. Ця таблиця зазвичай дуже маленька, не більше 30 записів. Тим самим усувається необхідність написання великих коментарів до коду – у великих системах це завдання досить трудомістке.

На першому етапі RacerX проходить по кожному файлу вихідного коду і складає граф логіки управління (Control Flow Graph, CFG). CFG містить інформацію про виклики функцій, загальної пам'яті, використання покажчиків і інші дані. При побудові графа цей інструмент також звертається до таблиці синхронізуючих примітивів і позначає виклики відповідних API-функцій.

Коли CFG повністю побудований, починається фаза аналізу, що включає перевірку на наявність гонок і взаємоблокувань. Перевірка графа CFG може займати багато часу, але для максимального прискорення цієї операції застосовуються методики редукції та кешування. Після перевірки логіки контекстів включається алгоритм набору блокувань (lockset), що виявляє потенційно можливі гонки. У ході аналізу взаємоблокувань для кожного захоплення блокування обчислюються цикли блокування.

Кінцева фаза передбачає пост-обробку всіх виявлених помилок з метою їх упорядкування за важливістю та критичністю. Автори програми чимало зробили для того, щоб максимально скоротити кількість помилкових позитивних

результатів і підвищити надійність виявлення помилок (що і слід очікувати від статичного аналізу). Результати роботи цього засобу дійсно вражають. Його практична користь в області тестування очевидна.

Chord [23], [61] – засіб статичного аналізу для Java, контекстно-залежний, але не чутливий до логіки управління. Незалежність від логіки управління забезпечує більш високу масштабованість в порівнянні з іншими засобами статичного аналізу, але за рахунок зниження точності. Це засіб також бере до уваги особливі синхронізуючі примітиви, доступні в Java. У ньому задіяний дуже складний алгоритм, для пояснення якого необхідно ознайомитися з великою кількістю концепцій.

Також можна згадати декілька вітчизняних розробок засобів статичного аналізу коду. Наприклад загальний підхід до аналізу структури програм реалізований у системі **V-Ray System**, що розробляється в лабораторії Паралельних інформаційних технологій НІВЦ МГУ і використовує в якості основи для аналізу програм параметричний опис графа алгоритму (інформаційної історії реалізації програм) [62]. Інший інструмент - **RRD** є засобом динамічної верифікації без запам'ятовування станів. Він призначений, перш за все, для тестування багатопоточних алгоритмів (алгоритмів синхронізації, багатопоточних структур даних і т.і.) [57].

ВИСНОВОК

Щоб задіяти всі переваги паралельних обчислень, потрібно вдосконалювати наявні та розробляти нові процедури розробки та тестування ПЗ. Паралельні програми складно розробляти і тестувати, тому що в них виникають помилки нового типу, пов'язані з паралелізмом. Тому і засоби, що використовуються для допомоги в тестуванні і верифікації, повинні бути орієнтовані на специфіку роботи саме паралельних програм.

Одним із напрямків розв'язання проблем пов'язаних з тестуванням паралельних програм є поява таких підходів, як тестування на основі моделі (model based testing) або на основі специфікацій (specification based testing).

Тестування на основі моделей дозволяє систематизувати й автоматизувати процес розробки тестових наборів за допомогою використання різного роду математичних моделей. І в тих випадках, коли невеликого ручного тестування виявляється недостатньо для забезпечення необхідного рівня якості програмного забезпечення, тестування на основі моделей дозволяє добитися істотного підвищення якості тестування з витратами меншими, ніж при використанні інших підходів.

Переваги тестування на основі моделей полягають в тому, що:

- тести на основі специфікації функціональних вимог більш ефективні, оскільки вони більшою мірою націлені на перевірку функціональності, ніж тести, побудовані тільки на знанні реалізації;
- на основі формальних специфікацій можна створювати тести, що перевіряються самі (self-checking), оскільки з формальних специфікацій часто можна витягти критерії перевірки результатів цільової системи.

В якості математичної моделі для аналізу та верифікації програм, було розглянуто апарат мереж Петрі. Мережі Петрі представляють собою апарат моделювання динамічних дискретних систем і є одним з найбільш адекватних способів опису асинхронного виконання паралельних процесів, у тому числі в розподілених системах.

Слід зазначити, що, оскільки мережі Петрі призначені більше для моделювання систем, представлення на їх базі архітектури проектованої програмної системи, у тому числі паралельної, іноді є дещо ускладненим. Це є наслідком не занадто високої узгодженості цієї моделі з популярними на даний момент методами програмування. Проте, незважаючи на це, вони є наймогутнішим засобом для побудови паралельних обчислень, що включають у себе можливості існуючих популярних засобів. Крім того, представлення обчислювального процесу у вигляді мережі Петрі може бути не прив'язане до самої реалізації її функціонування і може бути побудоване шляхом виділення підзадач в окремі функціональні блоки. На мережі Петрі в цій ситуації покладається лише задача організації послідовності їхнього виконання і синхронізація.

Список літератури

1. Карпов А. Тестирование параллельных программ. [Электронный документ]. Режим доступа: <http://www.viva64.com/ru/articles/parallel-programming/>. Проверено: 12 вересня 2010 р.
2. Кузнецов С. Кризис параллельного мира. [Электронный документ]. Режим доступа: <http://citforum.ru/computer/2009-12/>. Проверено: 12.09.2010.
3. Вирбицкайте И. Б. Формальные модели и анализ корректности параллельных систем и систем реального времени. Автореферат. – Красноярск, 2001. – 31 с.
4. Покозий Е. А. Методы спецификации и верификации параллельных моделей с непрерывным временем. Автореферат диссертации на соискание ученой степени кандидата физико-математических наук. – Новосибирск: Институт систем информатики Сибирского отделения РАН, 1999. – 30 с.
5. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация, применение. – СПб: БВХ. – 2003. – 104 с.
6. Воеводин Вл. В. Решение больших задач в распределённых вычислительных средах. // Автоматика и телемеханика. – 2007. – №5. – С. 32-45.
7. Вальковский В.А. Распараллеливание алгоритмов и программ: Структурный подход. – М.: Радио и связь, 1989. – 176 с.
8. Кларк Э. М., Грамберг О., Пелед Д. Верификация моделей программ. Model Checking. – М.: МЦНМО, 2002. – 416 с.
9. Голенков Е.А., Соколов А.С. Метод автоматического построения модели параллельной программы в терминах сетей Петри. // Вычислительные методы и программирование. – 2005. – С. 9.
10. Сеницын С.В., Налютин Н.Ю. Верификация программного обеспечения. – М.: БИНОМ. Лаборатория знаний, 2006. – 368 с.
11. Васильев В.В., Кузьмук В.В. Сети Петри, параллельные алгоритмы и модели мультипроцессорных систем – К.: Наукова думка, 1990 – 216 с.
12. Питерсон Дж. Теория сетей Петри и моделирования систем. – М.: Мир, 1984. – 264 с.
13. Котов В.Е. Сети Петри. М.: Наука. Главная редакция физико-математической литературы, 1984. – 160 с.
14. Sansifer R., Beaven M., Marinescu D.C. Modeling concurrent programs with colored Petri nets. Journal of System and Software. – 1994 p. – №2.
15. Sansifer R., Marinescu D.C. Petri net models of concurrent programs with colored Petri nets. Microelectronics and reliability. – 1991 p. – №4.
16. Myers J.G. The Art of Software Testing. Second Edition. – Hoboken, New Jersey.: John Wiley & Sons. – 2004p. – 255 с.
17. Майерс Г. Искусство тестирования программ. – М.: Финансы и статистика, 1982. – 174 с.
18. Что может предложить автоматизация тестирования? Или выгоды автоматизации тестирования. // Портал об автоматизированном тестировании ПО. [Электронный документ]. Режим доступа: <http://automated->

- testing.info/knowledgebase/article/chto-mozhet-predlozhit-avtomatizacija-testirovanija-ili-vygody-avtomatizacii-t. Перевірено: 11.09.2010.
19. Подход UniTest к разработке тестов: достижения и перспективы / [Баранцев А.В., Бурдонов И.Б., Демаков А.В., Зеленов С.В. та ін.]. [Електронний документ]. Режим доступу: <http://citforum.ru/SE/testing/unitesk/> Перевірено: 11.09.2010.
 20. Тестирование программного обеспечения. [Електрон.документ]. Режим доступу:http://ru.wikipedia.org/wiki/Тестирование_программного_обеспечения Перевірено: 11.09.2010.
 21. Тестирование по стратегии чёрного ящика. [Електронний документ]. Режим доступу: http://ru.wikipedia.org/wiki/Тестирование_по_стратегии_чёрного_ящика. Перевірено: 11.09.2010.
 22. Стратегия тестирования по принципу «Белого ящика». [Електронний документ]. Режим доступу: http://ru.wikipedia.org/wiki/Стратегия_тестирования_по_принципу_«Белого_ящика». Перевірено: 11.09.2010.
 23. Патил Р.В., Джордж Б. Инструменты и методики для выявления проблем, связанных с параллельным выполнением. [Електронний документ]. Режим доступу: <http://www.microsoft.com/Rus/Msdn/Magazine/2008/07/rasparallelivanie.mspx#Section3>. Перевірено: 11.09.2010.
 24. Статический анализ кода. [Електронний документ]. Режим доступу: http://ru.wikipedia.org/wiki/Статический_анализ_кода. Перевірено: 11.09.2010
 25. Динамический анализ кода. [Електронний документ]. Режим доступу: http://ru.wikipedia.org/wiki/Динамический_анализ_кода. Перевірено: 11.09.2010.
 26. Тестирование на основе моделей. / [Петренко А., Бритвина Е., Грошев С. и др.] // Открытые системы. – 2003. – [Електронний документ]. Режим доступу: <http://citforum.ru/SE/testing/model/>. – Перевірено: 13.09.2010.
 27. Model checking. [Електронний документ]. Режим доступу: http://en.wikipedia.org/wiki/Model_checking. Перевірено: 11.09.2010.
 28. Федотов И.Е. Некоторые приемы параллельного программирования: Учебное пособие. - М.: МГИРЭА(ТУ), 2008. – 188 с.
 29. Онищенко Б.О., Супруненко О.О. Управляючі мережі Петрі, як засіб моделювання та автоматизованого аналізу алгоритмічних конструкцій. // Вісник запорізького національного університету. – 2009.– № 1.– С.197-203.
 30. Голенков Е.А. Итоговый отчет о выполнении научного проекта. «Развитие технологии параллельного программирования на базе формализма сетей Петри». – Институт Автоматики и Процессов Управления ДВО РАН. – 2004. – 71 с.
 31. Проверка моделей. [Електронний документ]. Режим доступу: http://ru.wikipedia.org/wiki/Проверка_моделей. Перевірено: 11.09.2010.
 32. BLAST (статический анализатор). [Електронний документ]. Режим доступу: [http://ru.wikipedia.org/wiki/BLAST_\(статический_анализатор\)](http://ru.wikipedia.org/wiki/BLAST_(статический_анализатор)). Перевірено: 11.09.2010.

33. CADP: Construction and Analysis of Distributed Processes. [Електронний документ]. Режим доступу: <http://en.wikipedia.org/wiki/CADP>. Перевірено: 11.09.2010.
34. CHESS. [Електронний документ]. Режим доступу: [http://ru.wikipedia.org/wiki/CHESS_\(динамический_анализатор\)](http://ru.wikipedia.org/wiki/CHESS_(динамический_анализатор)). Перевірено: 11.09.2010.
35. CHESS: Find and Reproduce Heisenbugs in Concurrent Programs. [Електронний документ]. Режим доступу: <http://research.microsoft.com/en-us/projects/CHESS/>. Перевірено: 11.09.2010.
36. ISP Formal Verification Tool. [Електронний документ]. Режим доступу: http://en.wikipedia.org/wiki/ISP_Formal_Verification_Tool. Перевірено: 11.09.2010.
37. JPF. [Електронний документ]. Режим доступу: <http://javapathfinder.sourceforge.net/>. Перевірено: 11.09.2010.
38. Java PathFinder. [Електронний документ]. Режим доступу: http://ru.wikipedia.org/wiki/Java_PathFinder. Перевірено: 11.09.2010.
39. MoonWalker – model checking. NET applications. [Електронний документ]. Режим доступу: <http://wwwhome.cs.utwente.nl/~ruys/moonwalker>. Перевірено: 11.09.2010.
40. MRMC. [Електронний документ]. Режим доступу: <http://www.mrmc-tool.org/trac/>. Перевірено: 12.09.2010.
41. Markov Reward Model Checker (MRMC). [Електронний документ]. Режим доступу: [http://en.wikipedia.org/wiki/Markov_Reward_Model_Checker_\(MRMC\)](http://en.wikipedia.org/wiki/Markov_Reward_Model_Checker_(MRMC)). Перевірено: 12.09.2010.
42. NuSMV. [Електронний документ]. Режим доступу: <http://en.wikipedia.org/wiki/NuSMV>. Перевірено: 12.09.2010.
43. NuSMV: a new symbolic model checker. [Електронний документ]. Режим доступу: <http://nusmv.fbk.eu/>. Перевірено: 12.09.2010.
44. PRISM (model checker). [Електронний документ]. Режим доступу: [http://en.wikipedia.org/wiki/PRISM_\(model_checker\)](http://en.wikipedia.org/wiki/PRISM_(model_checker)). Перевірено: 12.09.2010.
45. PRISM. Офіційний сайт. [Електронний документ]. Режим доступу: <http://www.prismmodelchecker.org/>. Перевірено: 12.09.2010.
46. Rabbit Model Checker. [Електронний документ]. Режим доступу: http://en.wikipedia.org/wiki/Rabbit_Model_Checker. Перевірено: 12.09.2010.
47. Rabbit and Cottbus Timed Automata. Офіційний сайт. [Електронний документ]. Режим доступу: <http://www.sosy-lab.org/~dbeyer/Rabbit/>. Перевірено: 12.09.2010.
48. SPIN. [Електронний документ]. Режим доступу: <http://spinroot.com/spin/whatispin.html>. Перевірено: 11.09.2010р.
49. SPIN model checker. [Електронний документ]. Режим доступу: http://en.wikipedia.org/wiki/SPIN_model_checker. Перевірено: 11.09.2010.
50. Vereofy. [Електронний документ]. Режим доступу: <http://en.wikipedia.org/wiki/Vereofy>. Перевірено: 11.09.2010.
51. Vereofy / TU-Dresden. Офіційний сайт. [Електронний документ]. Режим доступу: <http://www.vereofy.de/>. Перевірено: 11.09.2010.

52. μ CRL. Офіційний сайт. [Електронний документ]. Режим доступу: <http://homepages.cwi.nl/~mcr1/>. Перевірено: 11.09.2010.
53. UPPAAL. Офіційний сайт. [Електронний документ]. Режим доступу: <http://www.uppaal.com/>. Перевірено: 11.09.2010.
54. Uppaal Model Checker. [Електронний документ]. Режим доступу: http://en.wikipedia.org/wiki/Uppaal_Model_Checker. Перевірено: 11.09.2010.
55. List of model checking tools. [Електронний документ]. Режим доступу: http://en.wikipedia.org/wiki/List_of_model_checking_tools. Перевірено: 11.09.2010.
56. Zing. Microsoft Research. [Електронний документ]. Режим доступу: <http://research.microsoft.com/en-us/projects/Zing/>. Перевірено: 11.09.2010.
57. Карпов, А. Интервью с Дмитрием Вьюковым – автором верификатора Relacy Race Detector (RRD). [Електронний документ]. Режим доступу: http://www.viva64.com/content/articles/code-analyzers/?f=Vyukov_rus.html&lang=ru&content=code-analyzers. Перевірено: 11.09.2010.
58. UniTESK. Офіційний сайт. [Електронний документ]. Режим доступу: <http://www.unitesk.ru/>. Перевірено: 13.09.2010.
59. Технология UniTESK. [Електронний документ]. Режим доступу: <http://ru.wikipedia.org/wiki/UniTesK>. Перевірено: 13.09.2010.
60. Intel® Thread Checker. Intel. [Електронний документ]. Режим доступу: <http://software.intel.com/en-us/intel-thread-checker/>. Перевірено: 13.09.2010.
61. Jchord. A Static and Dynamic Program Analysis Framework for Java. [Електронний документ]. Режим доступу: <http://code.google.com/p/jchord/>. Перевірено: 11.09.2010.
62. V-Ray. Офіційний сайт. [Електронний документ]. Режим доступу: <http://v-ray.parallel.ru/index.html>. Перевірено: 11.09.2010.
63. Питерсон Дж. Теория сетей Петри и моделирования систем. – М.: Мир, 1984. – 264 с.
64. BLAST: Berkeley Lazy Abstraction Software Verification Tool. MTC - Models and Theory of Computation. [Електронний документ]. Режим доступу: <http://mtc.epfl.ch/software-tools/blast/index-epfl.php>. Перевірено: 11.09.2010.
65. Карпов Ю.Г. Model Checking. Верификация параллельных и распределенных программных. СПб.: БХВ-Петербург, 2010. – 560 с.
66. Авербух, В.Л. Состояние дел в визуализации программного обеспечения параллельных вычислений / В.Л. Авербух, А.Ю. Байбалин, Д.Р. Исмагилов, А.Ю. Казанцев // 15-я Междунар. конф. по компьт. графике и её прил. : тр. конф.- ЗАО РИЦ "Прайс-курьер", 2005. – С.179-186.
67. Хорошилов А.В. Спецификация и тестирование систем с асинхронным интерфейсом. [Електронний документ]. Режим доступу: http://citforum.ru/SE/testing/asynchronous_interface/. Перевірено: 11.09.2010.

РАЗДЕЛ 2. Методы и средства моделирования и распознавания информативных участков в автоматических системах ириодиагностики с использованием параллелизма (Волошин М.В.)

Глаз – это уникальный орган, с помощью которого человек получает 80% всей окружающей его информации. Но помимо зрения у него имеются еще ряд функций. Речь идет об энергетической, светозащитной, диагностической и иммунологической функциях.

В процессе эмбрионального развития орган зрения формируется из тканей центральной нервной системы, а именно головного мозга. Анатомы говорят, что глаз – это вынесенный наружу мозг. Строение глаза и его структура несут в себе огромную информацию о человеке и его здоровье.

Ириодиагностика – это метод быстрой оценки состояния организма человека по признакам, которые наблюдаются на радужной оболочке глаза. Локальные особенности, именуемые иридологическими знаками (признаками) отражают, благодаря особым свойствам нервной системы человека, состояние практически всех органов и систем организма человека и используются для постановки диагноза с помощью, так называемых, схем (карт) проекционных зон (рис. 1).

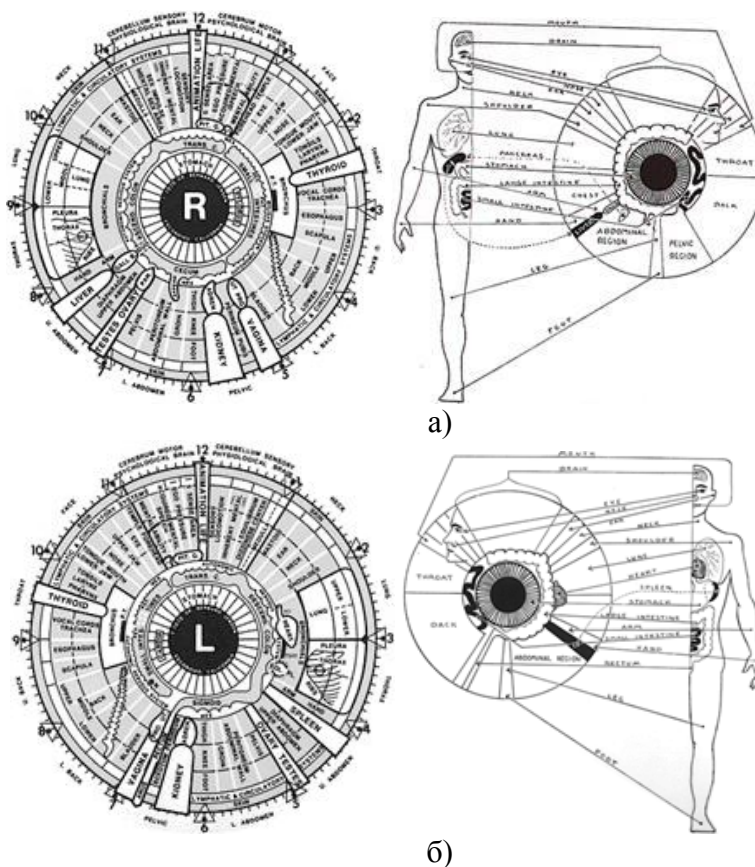


Рис. 1. Схемы проекционных зон правого (а) и левого (б) глаза

И пускай точно говорить о диагнозе нельзя, но зато можно выявлять заболевания на таких ранних стадиях, когда обычными методами диагностировать болезнь еще нельзя.

Особое значение в последние десятилетия приобретают иридогенетические исследования. В американском противораковом центре Бетесде иридографические методики применяются для дифференциальной диагностики доброкачественных новообразований. Одним из перспективных, и до сих пор мало разработанным направлением является вопрос о возможности воздействия узконаправленным лучом лазера на отдельные элементы радужки с лечебной целью, но пока эта идея интересна лишь в теоретическом отношении ввиду чрезвычайной сложности её методического осуществления, сомнительностью терапевтической эффективности и высокого риска развития непредсказуемых осложнений.

В нашей стране вопросами иридодиагностики стали серьёзно заниматься с 1967 г. Е.С. Вельховер и Ф.Н. Ромашов, а в 1984 г. решением Бюро президиума Учёного медицинского совета СССР метод иридодиагностика рекомендован к внедрению в широкую медицинскую практику.

2.1. История возникновения иридодиагностики

За полторы тысячи лет до нашей эры в древнем Египте правил фараон Тутанхамон. Правителя мучили непонятные недуги. И он призвал к себе жреца-врачевателя, чтобы тот установил причину его недомогания. Жрец принёс с собой несколько небольших сосудов, песочные часы и металлические пластины, покрытые особой серебристой жидкостью. Ел Акс (так звали жреца), тщательно осмотрел фараона и расспросил его о болезни. Затем он усадил Тутанхамона и, поднеся одну из пластин к его глазу на расстояние 2 см, перевернул песочные часы, которые были рассчитаны на 4 минуты. Фараон, не моргая, терпеливо ждал. Когда весь песок просыпался, Ел Акс убрал пластину и нанёс на неё какую-то жидкость.

И снова жрец поднёс пластину к глазам фараона, на этот раз на 30 секунд. Затем Ел Акс ушёл в свою лабораторию, и до сих пор никто не знает, какими составами он покрывал пластины, но после химической обработки на них появилось цветное изображение радужки фараона [1].

Эти своеобразные фотографии можно увидеть и сегодня – они хранятся в Ватикане. Яркие краски изображений глаз на металле сохранились до наших дней. Секрет же «цветного фотографирования» Ел Акса так и остался нераскрытым. А по уцелевшим изображениям радужки Тутанхамона, мы можем сегодня знать, что правитель древнего Египта был очень болезненным человеком.

Ел Акс описал диагностику по радужке на двух папирусах длиной 50 метров и шириной 1,5 метра. Они были найдены при раскопках в Гизе и хранятся в Вавилонской библиотеке. А знаменитый жрец оставил после себя славу не только иридодиагноста, но и популяризатора глазной диагностики. Благодаря ему, иридодиагностика распространилась из Египта в Вавилон, Тибет, Индокитай и другие регионы.

Современное возрождение иридодиагностики связано с именем доктора медицины Игнаца Пекцели (J.Peczeli, 1826-1907) из Будапештского предместья Егервара. С его именем связаны систематизация иридологических тестов и первые обоснования метода иридодиагностики. Пекцели ничего не знал о древней глазной диагностике, и одна из легенд гласит, что всё началось с необычного случая, произошедшего с 11-летним Игнацем. Однажды, гуляя в лесу, мальчик обнаружил совиное гнездо и попытался достать из него яйцо. Внезапно налетевшая сова вцепилась мальчику в руку. В завязавшейся борьбе Игнац сломал сове лапу и тотчас же на радужке птицы на стороне повреждённой лапы появилась вертикальная чёрная полоса. Происшествие в лесу оставило неизгладимое впечатление в памяти будущего доктора. Позднее, в период обучения в Венском университете и особенно во время работы в хирургическом госпитале, Пекцели стал наблюдать за изменениями радужки у людей при различных заболеваниях. Он обнаружил, что каждому участку тела или органу соответствует определённый сегмент в радужке. В результате многолетних исследований учёного появилась на свет первая в мире схема проекционных зон радужки, за что её автора по праву называют основоположником современной иридодиагностики.

В 1866 году Пекцели опубликовал книгу «Открытие в области природы и искусство лечения», в которой изложил принципы диагностики по радужке. Несколько позднее он напечатал «Руководство по изучению глазной

диагностики». Эпиграфом к этому труду служили слова: «Глаз не только зеркало души, но и зеркало тела». Древние медики весьма успешно проводили диагностику и оценку состояния организма по так называемым окнам тела – глазам, ушам, носу, ротовой полости и кожным покровам. Современными учёными установлено, что «окна тела», за счёт сосредоточенных в них наружных рецепторов, являются очень чувствительными посредниками между внешним миром и окружающей средой. Вместе они составляют сложную систему прямой и обратной связи, по которой сигналы поражения организма выносятся наружу, в проекционные зоны пяти органов чувств: радужки глаза, ушной раковины, кожи, слизистой носа и языка. Сейчас учёные и врачи всего мира проявляют большой интерес к поиску новых, безопасных и достаточно надёжных методов экспресс-диагностики заболеваний, с помощью которых можно было бы проводить массовые профилактические осмотры населения. Одним из таких новых, а, точнее, хорошо забытых старых методов, и является иридодиагностика – метод неспецифической диагностики заболеваний по изменениям структуры и цвета радужной оболочки глаза.

2.2. Основные положения иридодиагностики

Из курса анатомии известно, что глаза – это часть головного мозга. В организме все взаимосвязано, поэтому состояние внутренних органов отображается головным мозгом своеобразным языком знаков. Радужная оболочка – это как раз та часть глаз, на которой «пишется» вся летопись здоровья человека, начиная с самого рождения.

Если глаза светлые, это обозначает, что у данного человека от рождения сильный иммунитет. От рождения разный цвет глаз, например, один глаз карий, другой – зеленый – говорит, что их обладатель – неуравновешенный тип. Цвет глаз является основным показателем в определении типа конституции человека.

Надо заметить, что согласно иридологии, естественными цветами глаз являются только карий, голубой и их смесь. Прочие оттенки и тона, какими бы красивыми они ни были, свидетельствуют о наличии накопленных в организме токсинов. Но если обладатели даже самых небесно-голубых глаз будут питаться исключительно в закусочных, то их глаза превратятся в карие. И наоборот, карие глаза посветлеют, если человек перейдет на более здоровую пищу. Зеленый же не является естественным цветом. Его генетическая основа – голубой. К ней подмешана желтизна, свидетельствующая о функциональном нарушении. В идеале радужная оболочка должна быть абсолютно светлой, что указывает на отсутствие у ее обладателя проблем с пищеварением. Но если где-то что-то не так, цвет изменится на том участке, который связан с «заболевшей» системой организма. Вот основные моменты, которые необходимо учитывать для того, чтобы по цвету глаз определить состояние организма:

- голубая радужка означает склонность к болезням, связанным с повышенной кислотностью, таким, как артрит, ревматизм, астма и язва желудка.
- коричневая радужка свидетельствует о склонности к расстройствам системы пищеварения, например, гастроэнтериту, запорам, заболеваниям центральной нервной системы.

- радужка голубовато-коричневая (зеленая) говорит о том, что у ее обладателя повышена кислотность и токсичность, связанная прежде всего с нарушениями функций нервной и пищеварительной систем.
- смешанные оттенки радужки часто имеют голубую основу, что, следовательно, указывает на склонность к недугам, присущим голубой радужки.

Плотность радужки:

- плотная радужка напоминает глаз как у детской куклы: ровная, гладкая, равномерно пигментированная, без пятен. Если на глазах радужка плотная, это признак отличной наследственности, выносливости, крепкого иммунитета. После болезней и операций здоровье, скорее всего, восстанавливается быстро и полностью. Велика вероятность дожить до 80-85 лет, а может, и до 90;
- рыхлая радужка говорит о «средней» наследственности. При больших психических и физических нагрузках у таких людей возможны нервные срывы, чрезмерная раздражительность, головные и сердечные боли, спазмы различных органов, депрессия. Но если ритм жизни умеренный, если человек не «бросает свое здоровье на произвол судьбы», то без особых проблем можно дожить минимум до 75-80 лет;
- очень рыхлая радужка, в которой волокна расщеплены и много «дыр», – признак слабого иммунитета, невысокой степени выносливости. Даже при незначительных нагрузках или стрессах возможны нервные срывы и возникновение заболеваний. Но тем не менее есть шансы дожить до 70-75 лет.

Цветные рисунки:

- вокруг зрачка желтоватая окраска – признак зашлакованности кишечника, печени, желчного пузыря;
- белая дуга по краю радужки – верный признак атеросклероза. Если в верхней ее части атеросклероз мозговых сосудов, в нижней – сосудов ног;
- полукольца или кольца, проходящие через всю поверхность радужки, вокруг зрачка, обозначают, что обладатель такой радужки человек чувствительный, но держащий в себе отрицательные эмоции, обиды, напряжение. От этого психического напряжения страдают нервная и сердечно-сосудистая системы.
- пятна говорят о нарушениях в конкретных органах. Для того чтобы определить, где находится болезнь, нужно посмотреть на схему проекций внутренних органов и определить, в каком органе эти пятна находятся.

2.3. Иридосхемы

В настоящее время известно более тридцати схем проекционных зон радужной оболочки глаза, используемых в иридодиагностике, и их варибельность способна послужить предметом особого углублённого исследования (рис. 2). Например, проекцию щитовидной железы можно найти только на схемах В. Jensen и А. Castro, проекцию тимуса – у К. Gunter и

J. Angerer, солнечного сплетения – у В. Jensen и F. Roberts, воротной вены – у П. Димкова и G. Jausas. Наиболее популярной среди практикующих иридологов стала большая группа германо-американских схем с вертикальным принципом проецирования. Их авторы – J. Deck (1954), J. Angerer (1957), K. Gunter (1959), F. Roberts (1962), В. Jensen (1964), A. Castro (1979). Но существует и иной, эмбриологический принцип отображения тела человека на радужной оболочке. Его придерживаются французские авторы R. Bourdiol (1975), G. Jausas (1983). Противоречие между этими двумя школами заключается в том, что сторонники «вертикального принципа», основываясь на идее раннего гастроло-сегментарного периода эмбрионального формирования путей и центров иридо-диэнцефального отображения, согласно которому зародыш человека, развиваясь из яйцеклетки, проходит стадию гастролы, на основе которой в результате деления и усложнения формируются отдельные органы и системы, располагают проекцию желудочно-кишечного тракта в зрачковой зоне, проекции головного мозга в верхней, а нижних конечностей – в нижней части ириса. Оппонируя им, апологеты «эмбриологического типа» таким образом описывают соматотопию радужной оболочки, что расположение проекций органов напоминает расположение 6-7-недельного эмбриона в полости матки, причём зрачковый пояс остаётся висцероинтактным, органы желудочно-кишечного тракта проецируются в нижние отделы цилиарного пояса, а проекции головы и нижних конечностей занимают верхнюю его часть.

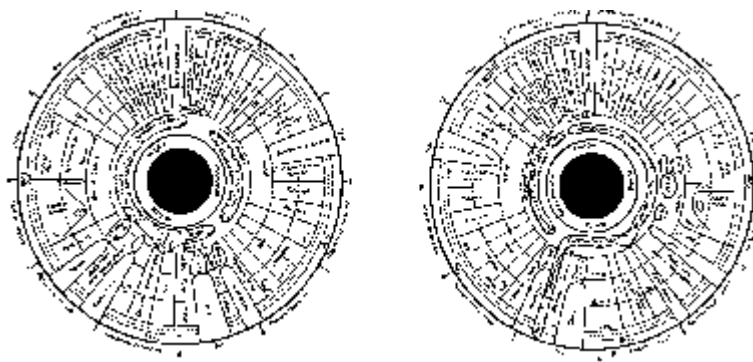


Рис. 2. Интегративная схема иридологических проекционных зон

Таким образом, соматотопия радужной оболочки глаза подразумевает сложную архитектуру зон проецирования. По установившемуся мнению она подчиняется двум правилам топографического деления – радиальному и концентрическому. Радиальному подчиняется область цилиарного пояса, и индикация сегментов проводится по угловому (в градусах) или циферблатному (в часах и минутах) принципам; концентрическое деление действует на территории как зрачкового, так и цилиарного поясов и состоит из девяти кольцевых зон. Одним из сложнейших вопросов иридогеографии остаётся геометрия различных типов межорганных соотношений, как сосуществующих в пределах одного ириса, так и с учётом энантиморфных сопряжений. К моноиридальным обычно относят симметричные связи бронхов, гениталий и почек, а так же сагиттальные связи головы и туловища и диагональные связи, соединяющие заднюю и переднюю поверхности корпуса с шеей и головой; бирадиальными связями принято считать зеркальные отображения головного мозга, лёгких, кишечника, почек и др. [1]. Но

даже у следующих этим общим правилам авторов встречаются подчас существенные разногласия. Например, зону проекции головного мозга J. Deck, A. Castro, B. Jensen, F. Roberts, П. Димков располагают в цилиарном поясе с 11:00 до 01:00 часа, K. Gunter расширяет её латерально на 10 мин. за счёт проекции продолговатого мозга, а J. Angerer – на 20 мин., но уже за счёт височной доли и мозжечка, G. Jausas и R. Bourdiol, напротив, укорачивают её с медиальной стороны на 30 мин.; местоположение проекции гипофиза так же трактуется крайне разноречиво, как и местоположение зоны надпочечника в нижней части цилиарного пояса, которую представители разных школ смещают медиальнее или латеральнее 6-часового радиуса. Сомнительной представляется и проекция позвоночного столба в нижнемедиальном квадранте, более правдоподобной выглядит её отображение на автономном кольце радужки.

Итак, очевидно, что и до нынешнего времени многое в иридологии нуждается в дальнейшей верификации и строжайшей научной проверке с использованием современных технологий, но поразительная жизнеспособность научной идеи соматотопического проецирования, подводящая нас сегодня к созданию концепции «голографического человека» вызывает несомненное уважение.

2.4. Возможности иридодиагностики

С помощью иридодиагностики можно сделать многое, но есть также вещи, которые она не может сделать. Итак, необходимо знать ограничения и возможности иридодиагностики, чтобы можно было ее лучше использовать в качестве инструмента. Иногда люди используют иридодиагностику, чтобы подтвердить то, что сказал врач. Иногда используют иридодиагностику, прежде чем пойти к врачу, чтобы иметь представление о состоянии своего организма. Иридодиагностика может выявить конституционные сильные и слабые стороны, унаследованные сильные и слабые стороны, общий уровень здоровья, пищевой и химической потребности, расположение токсинов в организме, диетические потребности и проблемы, качество нервной системы и внутренних органов, реакция пациента на лечение, уровень кислотности в организме и т.д.. Иридодиагностика также уникальна тем, что она может «видеть» организм в целом и не требуется для этого много различных тестов. Когда вы смотрите на глаз человека – вы смотрите на все тело сразу. Вы видите, таким образом, как различные части тела, влияют друг на друга.

Есть много вещей, которые иридодиагностика не может делать. Во-первых, иридодиагностика не делает диагностику заболеваний. Радужка может только сказать вам, где проблемы, но насколько серьезными они являются – она не может сказать. Иридодиагностика также не раскрывает информацию о теле во время операций. Это потому что во время анестезии в нервных импульсах происходит «короткое замыкание». Это означает, что глаза будут иногда «замораживать» чтение некоторых участков тела.

Иридодиагностика не показывает нормального состояния тела. Вы не можете сказать женщине, что она беременна по рисунку диафрагмы, так как это «нормальное» состояние организма. Например, если человек родился без одной почки – это тоже не возможно «зарегистрировать» в глазах, так как это «нормально» для этого человека.

Иридодиагностика не экстрасенс. Вы не можете прочитать чьи-нибудь мысли используя иридодиагностику. Тем не менее, используя иридодиагностику можно раскрыть черты и призывы личности, что является достаточно близким к чтению мыслей. Приложив немного знаний из китайской медицины можно много, чего узнать о личности человека.

Иридодиагностика не может сказать вам, что авария могла произойти с человеком, но она может показать убытки от аварии. Например, можно увидеть сломанную руку в радужной оболочке глаза, но не можно сказать, где она была сломана – на лыжной прогулке или в автомобильной аварии.

Иридодиагностика не может показать, вирусы, паразиты или микробы, но она может показать условия, которые могут способствовать этому.

2.5. Современное состояние иридодиагностики на Украине

В Украине в настоящее время курс иридодиагностики преподается в Киевском медицинском институте Украинской ассоциации народной медицины (УАНМ). Заведующая курсом иридодиагностики, кандидат биологических наук, доцент Тамила Андреевна Сядро – ученик и соратник В.В.Кривенко, давний энтузиаст компьютерной иридодиагностики [2]. Еще в конце 80-х годов она участвовала в разработке первых украинских компьютерных иридодиагностических программ. В 1998 году, в Хорватии опубликована монография Т.А.Сядро «Иридодиагностика». В 1995-96 гг. Т.А.Сядро выполняла роль эксперта при разработке компьютерной иридодиагностической программы ЭСИД-2, а в 2000-2001гг. участвовала в разработке значительно более совершенной программы ЭСИД-3 [3].

Современная процедура проводится с помощью цифрового фотоаппарата и компьютера, радужка фотографируется и изображение выводится на монитор, после чего анализируется и обрабатывается при посредстве иридологических программ, а также электронных атласов по иридодиагностике разработанных сотрудниками кафедры акустики и акустоэлектроники факультета электроники НТУУ («КПИ») и медицинского института Украинской ассоциации народной медицины.

Преимущества иридодиагностики перед другими методами исследования человеческого организма:

- метод иридодиагностики абсолютно безболезненный и безвредный, в отличие от многих широко применяемых методик обследования, не имеет никаких противопоказаний для применения (со стороны общего состояния), не требует предварительной специальной подготовки обследуемого и в большинстве случаев позволяет дать диагностическое заключение сразу после проведенного осмотра;
- иридодиагностика может выявлять заболевания на ранних стадиях, когда обычными методами диагностировать болезнь ещё нельзя;
- с помощью иридодиагностики можно обнаружить изменения во всех органах и локализовать их с достаточной степенью точности, а также судить о конституциональных особенностях организма, об общем состоянии нервной, сосудистой систем, системы микроциркуляции, о стрессовой готовности организма и пр.;

- иридодиагностика даёт возможность оценить – вызвано ли заболевание каким-либо поражением органов, или связано с накоплением в организме токсинов, шлаков, возникло из-за повышенных нервных нагрузок и т.п., и дать соответствующие рекомендации по оздоровлению.

Приведенный перечень довольно убедителен, чтобы рекомендовать иридодиагностику не только как эффективный метод профилактических скрининг-обследований взрослого населения, но и как один из основных методов мониторинга состояния здоровья школьников.

В настоящее время в школах Украины метод иридодиагностики применяется, однако в весьма ограниченных объемах. Объяснить это можно двумя причинами. Во-первых, среди медицинских специалистов распространено мнение (вполне справедливое), что для гарантированно высокого качества иридодиагностики иридолог должен иметь квалификацию врача-терапевта. Однако средние школы Украины могут разрешить себе иметь специалистов только со средним медицинским образованием. Во-вторых, проведение иридодиагностики обычно требует специального дорогого оборудования (иридоскоп или щелевая лампа, а в последнее время – цифровая фотокамера или видеокамера), приобрести которое не по силам большинству средних школ Украины.

Вместе с тем, анализ мирового опыта показывает, что указанные трудности можно успешно преодолеть, если использовать для проведения иридодиагностики специальные информационно-справочные или экспертные системы. С помощью таких компьютерных иридодиагностичных систем высококачественное заключение иридолога может быть подготовлено медицинским специалистом средней квалификации, имеющимся почти в каждой средней школе Украины. А вместо иридоскопа или щелевой лампы можно использовать даже обычную пленочную фотокамеру, фотографируя радужку глаза в режиме макросъемки.

Такие выводы подтверждаются и собственным опытом специалистов Киевского Финансового лицея, которые, начиная с 1998 года, апробируют ежегодное иридологическое обследование лицеистов с применением компьютерной программы ESID-2. С 2000 года профилактические обследования школьников проводятся с использованием более мощной компьютерной программы для иридодиагностики – экспертной системы ESID-3. Наконец, с 2002 года, после приобретения цифровой фотокамеры и компьютерной программы-фоторедактора GIDRA, стало возможным говорить о создании в Финансовом лицее собственного АРМ'а иридолога.

Приведенный пример свидетельствует о том, что современный уровень развития информационных технологий сделал принципиально возможным создание АРМ'а иридолога силами специалистов даже небольших медицинских центров и средних школ.

2.6. Перспективы развития

Перспективным направлением развития компьютерной иридодиагностики является ввод и обработка цветных изображений радужных оболочек глаз с применением цифровых фотокамер. Действительно, иридологу удобно иметь

перед глазами одновременно и цветное изображение радужки, и элементы интерфейса компьютерной диагностической программы, вместо того чтобы метаться между щелевой лампой (иридоскопом) и компьютером.

Однако здесь есть два существенно ограничивающих фактора: цена фотокамеры и качество получаемого цветного изображения.

Качество снимков можно охарактеризовать двумя показателями – разрешающей способностью и цветопередачей. Как показали эксперименты с цифровыми фотокамерами, приемлемое разрешение достигается для камер с матрицами светочувствительных элементов не менее 1024x768. Качество цветопередачи также зависит от выбора камеры и в ряде случаев может считаться вполне удовлетворительным.

Средняя цена бытовых цифровых фотокамер в 2000 году была близкой \$1000. Например, цена камеры MVC-FD 91 в январе 2000 года составляла \$1100. С течением времени наблюдается постепенное улучшение показателя цена / качество. Так, в 2011 году стоимость достаточно качественных бытовых фотокамер близка \$500 при количестве пикселей приблизительно 10 млн.

Таким образом, сегодня главным сдерживающим фактором применения цифровых фотокамер в иридодиагностике является их цена.

Автоматическая обработка цветных изображений радужных оболочек глаз – задача исключительно заманчивая, но и вместе с тем исключительно трудная. Попытки подобного рода известны. Например, в Московском Государственном центре «Контакт» под научным руководством Е.С.Вельхова разработана система АИК-01, где ввод изображений в компьютер производится с применением телевизионной видеокамеры, после чего обработка изображений осуществляется с помощью специальных программ (впоследствии была разработана система АИК-02).

Вместе с тем, есть основания сомневаться в эффективности технического решения, основанного на применении телевизионных видеокамер – как показывают экспериментальные исследования, их разрешающая способность недостаточна для получения качественных изображений.

В 2001 году в Интернете появилось сообщение о канадской компьютерной программе Vexel IRINA Version 2.0 стоимостью \$2250 (канадских долларов). Эта программа сертифицирована почему-то в Корее и, если верить рекламной информации, обеспечивает 80%-ю достоверность диагностики.

Однако доверять рекламной информации опасно. Весьма значительны трудности, с которыми встречаются разработчики соответствующего программного обеспечения. Одна из главных трудностей – различные артефакты типа бликов от вспышки, отражений век и ресниц, рефракции из-за слезы. Однако даже если бы не было этих помех, анализ цветного изображения радужки все равно представляет сложную задачу даже для опытных иридологов, не говоря уж о специалистах средней и низкой квалификации. Поэтому при автоматическом анализе цветных изображений радужки следует ожидать значительного процента «брака» в виде неверных или неточных выводов. Как следствие, иридолог вынужден будет внимательно проверять все автоматически сделанные выводы. Затраты времени и усилий на такую проверку в конечном счете не окупаются.

Диагностика на расстоянии с помощью Интернета. В связи с быстрым развитием сети Интернет естественно возникает вопрос об использовании

возможностей удаленного доступа к информации в задаче компьютерной иридодиагностики.

В литературе описаны некоторые попытки отечественных специалистов наладить контакты с зарубежными медиками. Несмотря на принципиальную техническую разрешимость задачи обмена информацией, оказалось, что не следует ожидать быстрого повсеместного внедрения технологии удаленного доступа на рабочие места врачей. Первая трудность, с которой встретились врачи – это «нестыковка» мнений специалистов из-за различия школ, к которым они принадлежат.

Есть и другие трудности технического, организационного и экономического характера. Поэтому нам представляется, что наибольшее распространение сетевые технологии могут найти в рамках локальных сетей.

2.7. Требования к ПО АРМ'а иридолога

Требования к ПО АРМ'а иридолога можно сформулировать следующим образом [4]:

- возможность эксплуатации иридологами различного уровня квалификации – от начинающих иридологов до высокопрофессиональных специалистов;
- наличие хорошо структурированных баз знаний;
- наличие простых и эффективных механизмов логического вывода;
- наличие возможности ввода / вывода текстовой, графической и, возможно, речевой информации в процессе диагностики либо в процессе обучения иридодиагностике;
- наличие удобного графического интерфейса пользователя.

2.7.1. Учет уровня профессиональной подготовки

Выделим два аспекта этого вопроса. Первый достаточно тривиален – существуют иридологи начинающие и опытные. Для начинающих иридологов компьютерная система нужна в первую очередь как электронный справочник, позволяющий восполнить отсутствие необходимых знаний. Требования опытных иридологов иные – они нуждаются главным образом в специализированном инструменте для быстрого формирования, редактирования и выдачи на печать текстов заключений.

Второй аспект менее очевиден и поэтому до настоящего времени не учтен в известных нам иридодиагностических системах [5]. Сущность его в том, что практикующих иридологов можно различать в соответствии с их мировоззрением. С одной стороны, это сторонники так называемой альтернативной медицины, исповедующих принципы «холизма», т.е. видения организма человека как единого целого – зачастую такие специалисты не имеют образования в объеме медицинского вуза европейского типа. С другой стороны, это специалисты с «традиционным» образованием в объеме медицинского вуза.

Для одновременного удовлетворения требований представителей этих двух групп процесс иридодиагностики должен быть организован в соответствии со схемой показанной на рис. 3.

В основе этой схемы – высказанная в [6] идея о желательности наличия у иридолога квалификации терапевта: при этом одно и то же лицо является носителем взаимно дополняющих концепций – синтетической (холистической) и аналитической.

Вместе с тем, следует отметить некоторую условность схемы на рис. 3: при реальной иридодиагностике последовательность этапов может быть иной, причем многие этапы могут выполняться параллельно. Достаточно условны и указанные на схеме обратные связи – на самом деле их намного больше, они охватывают и каждый этап в отдельности, и различные группы этапов, что адекватно отражает сложность логических рассуждений высококвалифицированного иридолога.

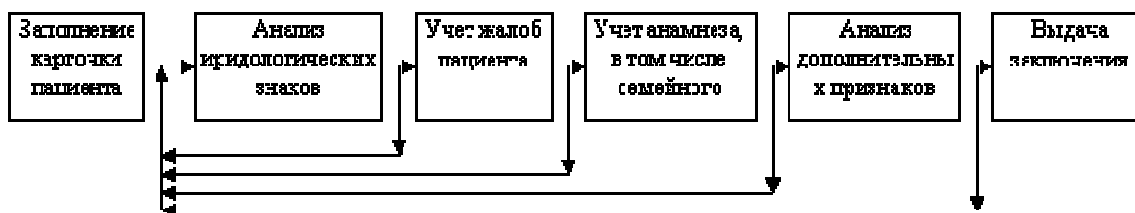


Рис. 3. Организация процесса иридодиагностики.

2.7.2. База знаний

Обобщенная структура БЗ, являющейся моделью памяти иридолога, представлена на рис. 4. Такая база знаний может содержать перечни не только иридологических, но и терапевтических сведений – в виде дополнительных, не иридологических признаков, соответствующих им патологий, а также рекомендаций по соответствующим профилактическим мероприятиям и лечению. При этом множества как признаков, так и рекомендаций по лечению, обычно группируются с использованием некоторых классификационных признаков.

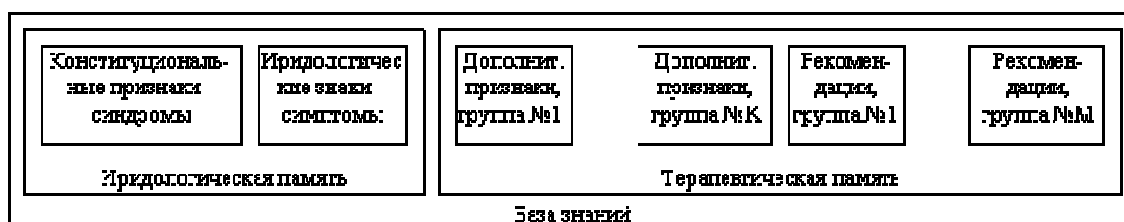


Рис. 4. Обобщенная структура базы знаний.

Так, например, иридологические признаки можно подразделить на синдромы (комплексные, конституциональные признаки) и симптомы (элементарные признаки) – такая группировка признаков сложилась в процессе развития иридологии и может быть объяснена несколькими причинами. Во-первых, это отражение иерархического характера организации знаний специалистами разной квалификации: известно, что высококвалифицированные специалисты оперируют более крупными блоками информации, нежели специалисты более низкой квалификации [5]. Другую причину деления признаков на элементарные и комплексные следует искать, по-видимому, в физиологических особенностях организма человека, внешне проявляющихся в виде статистической зависимости отдельных групп элементарных признаков.

2.7.3. Логический вывод

Отметим, что собственно логическому выводу предшествует поиск информации в прямом (признак \rightarrow заболевание) или/и обратном (заболевание \leftarrow признак) направлениях. Поясняется это тем, что для общей диагностики доминирующую роль играет поиск в прямом направлении, а для частной диагностики – в обратном [5].

Подразумевая под логическим выводом объединение информации о патологиях при задании нескольких признаков, отметим, что выбор механизма (алгоритма) логического вывода, существенно зависит от профессиональной ориентации пользователя. Так, для практикующих иридологов достаточен механизм логического суммирования с последующей сортировкой патологий, примененный в программе ESID-2 [5], а затем и в последующей программе ESID-3 [7], и описываемый соотношением:

$$Z = P_1\{P_2\{Y_i\}\},$$

где Z – заключение иридолога, представляющее собой множество наименований патологий, обнаруженных у пациента; $Y_i, i=1,2,\dots,K$ – i -тое подмножество наименований патологий, соответствующее i -тому наблюдаемому признаку X_i ; P_1 – оператор разбиения (сортировки) текста заключения на разделы, соответствующие системам организма (сердечно-сосудистая, дыхательная, желудочно-кишечная и т.д.); P_2 – оператор сортировки наименований патологий внутри каждого из разделов (по степени выраженности патологий).

Для иридологов-исследователей представляют интерес вероятностные механизмы формирования заключения с элементами Байесовской стратегии, подобные применявшиеся в программе ESID [5]. Апостериорная вероятность заболевания $Y_j, j=1,\dots,K, J_i$ для совокупности наблюдаемых признаков $\{X_1, K, X_l\}$ при этом может вычисляться по рекуррентной формуле:

$$P(Y_j / X_1, K, X_l) = \frac{l-1}{l} P(Y_j / X_1, K, X_{l-1}) + \frac{1}{l} P(Y_j / X_l),$$

где $P(Y_j / X_i)$ – апостериорная вероятность заболевания Y_j для каждого i -того признака X_i , вычисляемая в соответствии с выражением:

$$P(Y_j / X_i) = \begin{cases} \frac{1}{J_i} (1 - \frac{r}{R}) + \frac{r}{R} \frac{P(X_i / Y_j)}{\sum_j P(X_i / Y_j)}, & J_i > 1 \\ \frac{r}{R} P(X_i / Y_j), & J_i = 1 \end{cases}$$

где r – числовой коэффициент ($r=0$ K R), характеризующий выраженность наблюдаемого признака X_i .

Достаточно традиционным [4] для конституциональной и частной иридодиагностики является детерминистский механизм логического вывода путем суммирования баллов:

$$S_i = S_{i-1} + B_i,$$

где S_i – сумма баллов при задании i -го признака ($S_0 = 0$); B_i – количество баллов, соответствующее i -му признаку.

2.7.4. Ввод-вывод информации

В современных иридологических системах применяются такие формы ввода информации:

- с помощью клавиатуры или мыши (управление, формирование заключения);
- с выхода цифровой фотокамеры/видеокамеры (ввод изображения в компьютер).

Наиболее актуальным является ввод в память компьютера изображения радужной оболочки глаза – это не только быстрое архивирование изображения, но и значительное снижение утомляемости как пациента, так и врача за счет сокращения времени их контакта со специальной светооптической аппаратурой (щелевой лампой, иридоскопом).

Не менее важен и высококачественный вывод информации:

- на экран одного или нескольких мониторов;
- распечатка на цветном принтере;
- озвучивание текста.

Для иридолога первичным и потому приоритетным является вывод изображения на экран монитора – отсюда с необходимостью следуют и высокие требования к качеству монитора.

Вывод текста заключения на печать – также вполне понятная и очевидная процедура. Вывод на печать цветных изображений радужки глаза – задача, в принципе, легко решаемая с помощью цветных струйных принтеров. Наконец, голосовой вывод информации целесообразен при обучении иридологов с помощью электронных атласов. При этом иридолог может рассматривать изображение ириса на мониторе компьютера, одновременно слушая синтезированный голос с комментариями данного изображения.

Укрупненная структура компьютерной иридодиагностической программы [4] показана на рис. 5.

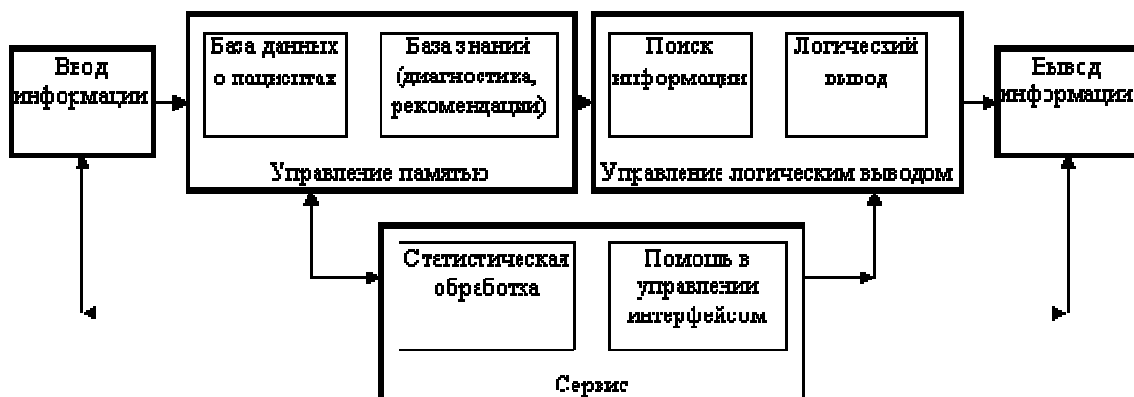


Рис. 5. Структура компьютерной иридодиагностической программы

Наличие в этой структуре базы знаний и механизма логического вывода позволяет отнести такую компьютерную программу к классу экспертных систем продукционного типа [5].

2.8. Компьютерные программы и рабочие станции для иридодиагностики

2.8.1. Компьютерные программы

К таковым, в первую очередь, отнесем компьютерные программы для быстрого автоматизированного формирования заключения иридолога на основании информации о наличии иридознаков на радужке глаза. Анализ иридознаков может производиться либо иридологом, либо полуавтоматически.

Наибольшее распространение получили диагностические компьютерные программы, с помощью которых иридолог анализирует иридознаки, наблюдаемые на радужной оболочке глаза. Такое наблюдение производят с помощью любых технических средств – лупа, иридоскоп, щелевая лампа, аналоговая или цифровая видеокамера, пленочная или цифровая фотокамера. Результаты наблюдений являются исходными данными для компьютерной программы, экспертный блок которой автоматически обрабатывает введенные врачом иридопризнаки и выдает заключение по результатам проведенного обследования. Примеры компьютерных программ такого рода: 1) «Скрининг-Д», «Радуга» и др. (Россия: Москва, ЗАО РИНК); 2) «ESID-3» (Украина: А.Продеус, Т.Сядро, КПИ-УАНМ, Киев); 3) «SoftIris Pro» (США: M. S. Abramov, Homewood, Illinois); 4) «Iris Camera» (США: Instituto Prana); 5) «IriSearch Pro» (Австралия: Roo du’Jardin, Naturopathic Technologies); 6) «Iridology Control Centre» (Канада: Bryan K. Marcia, Canadian Neuro-Optic Research Institute) [1].

Ко второй, более редкой, разновидности относятся компьютерные программы с элементами автоматизированного анализа цветных изображений радужки и оценки иридопризнаков, с последующим их анализом в экспертном блоке. К таким программам, например, относятся: 1) «Иридоанализ» (Россия: Москва, ЗАО РИНК); 2) «ИРИС» (Россия: Москва, ЦНИИ «Комета») [8]. Существенный недостаток таких программ – их дороговизна (примерно на порядок выше стоимости «обычных» компьютерных иридологических программ).

В особую группу можно выделить программы-утилиты для решения вспомогательных задач (например, для наложения прозрачных карт-транспарантов на изображения радужки с последующим выводом о принадлежности наблюдаемого иридознака той или иной проекционной зоне, для вспомогательной обработки полученных изображений радужки). Примером программ такого рода являются: 1) «SoftIris Chart» (США: M.S. Abramov, Homewood, Illinois); 2) «IriSearch Plus» (Австралия: Roo du’Jardin, Naturopathic Technologies) [8].

Компьютерная программа «Иридоанализ» ЗАО РИНК. Это Windows-приложение, разработанное в ЗАО РИНК (Россия: Москва). Программа ориентирована на запросы врача (терапевта, гомеопата, невропатолога, рефлексотерапевта и др.), практикующего иридоанализ как прогностический метод контроля проводимых врачом лечебно-оздоровительных мероприятий с

учетом конституциональных особенностей организма пациента, предрасположенности к патологии определенных органов, наследственности. Данная программа может быть рекомендована как опытным, так и начинающим иридологам.

Программа обеспечивает:

- ввод и отображение всей информации на мониторе;
- обработку и документирование цветных изображений радужек;
- работу с базой данных;
- альтернативное использование встроенной экспертной системы и составленного пользователем списка патологий органов и систем для формирования заключения с учетом результатов иридологического анализа;
- сопоставимость результатов обследований, проведенных в разное время в течение времени наблюдения пациента;
- автоматическую обработку изображений зрачка (децентрация, уплощения, деформации), автономного кольца с сохранением информации в базе данных.

Программа позволяет провести в диалоговом режиме работу по иридопризнакам по органам и системам.

Компьютерная программа «Скрининг-Д» ЗАО РИНК. Это Windows-приложение, разработанное в ЗАО РИНК. Предусматривает ведение базы данных, регистрацию пациента, ввод изображения радужной оболочки глаза, работу по иридопризнакам с изображениями зрачка, зрачкового пояса, автономного кольца, цилиарного пояса, а также проверку на сахарный диабет, риск инфаркта миокарда, гипертонию, онконастороженность, гинекологию, мастопатию, травмы черепа и нижних конечностей, повышенную эмоциональную лабильность, истощение нервной системы, риск невротических реакций. Экспертный блок автоматически обрабатывает введенные врачом иридопризнаки и выдает заключение по проведенному обследованию.

«Скрининг-Д» является «усеченной» версией программы «Иридоанализ» ЗАО РИНК.

Компьютерные программы для частной иридодиагностики ЗАО РИНК. Это четыре DOS-приложения, разработанные в ЗАО РИНК: «Экспертная система для иридодиагностики желудочно-кишечного тракта», «Ранняя диагностика сахарного диабета», «Экспертная система иридодиагностики заболеваний печени и желчевыводящих протоков», «Экспертная система для иридодиагностики сердечно-сосудистой системы». Данные программы работают в диалоговом режиме, имеют паспортную часть, анализ соответствующих зон радужной оболочки. Имеются рекомендации фитотерапевтического и гомеопатического лечения, а также справочник расположения проекционных зон обследуемых органов. Работа заканчивается выводом информации на принтер или монитор.

Недостаток программ – необходимость работы в DOS.

Компьютерная программа «Иридодиагностика» медицинского центра «Ибн Сина». Это разработка медицинского центра «Ибн Сина» фирмы «Новые системы» (г.Тамбов, Россия), предназначенная для проведения массовых профилактических осмотров, а также для индивидуального обследования пациентов с последующей статистической обработкой материала. При повторном

обследовании можно сравнивать иридологические карты, т.е. осуществлять динамические наблюдения (мониторинг).

К недостаткам программы «Иридодиагностика» относятся:

- время ввода информации в ЭВМ достигает 1,5 часов;
- ввод информации в ЭВМ осуществляется посредством предварительного заполнения технических носителей информации и может производиться только с помощью инженерного персонала.

Приведенная информация свидетельствует о моральной устарелости (на сегодняшний день) принятых технических решений.

Компьютерная программа «АРМ иридолога» фирмы «Арника монтана». Это разработка тамбовской фирмы «Арника монтана». Данная программа эксплуатируется на IBM-совместимых компьютерах, позволяет решать задачи массового профилактического и глубокого индивидуального осмотров. Помимо режимов ввода информации о наличии топических и тополабильных признаков, а также режима автоматизированного формирования заключения, имеет режим управления электронной картотекой пациентов и режим статистической обработки данных картотеки.

Существенным недостатком программы является неудобство ввода информации в компьютер: меню признаков реализовано в виде «бумажной» таблицы и аналогичной схемы проекционных зон, что нарушает целостность технологии компьютерной обработки и хранения информации.

Компьютерная программа ESID Киевского НПО «Славутич». Компьютерная программа ESID (MS DOS) разработана в 1989-91гг. в Киевском НПО «Славутич» и представляет собой небольшую экспертную систему продукционного типа, позволяющую осуществлять общую, дифференциальную и частную иридодиагностику с использованием элементов Байесовской стратегии [9], [5].

Программа ESID многоцелевая: она может использоваться как для практической иридодиагностики, так и для научных исследований и обучения иридологов.

В перечень ее возможностей входят:

- количественная оценка достоверностей возможных заболеваний;
- решение задач общей, дифференциальной и частной диагностики с поиском информации по схемам «признак-заболевание» и «заболевание-признак»;
- использование, наряду с иридодиагностическими, дополнительных признаков;
- аргументирование диагноза;
- автоматизированное формирование заключения врача с рекомендациями по лечению выявленных заболеваний;
- протоколирование результатов обследований, с возможностью последующей автоматизированной статистической обработки протоколов;
- модернизация базы знаний без привлечения профессиональных программистов;
- возможность автоматизированного учета результатов статистической обработки протоколов при модернизации базы знаний;

- количество распознаваемых нозологических форм – около 300;
- количество используемых диагностических признаков – порядка 1000;
- наличие режима «Помощь» в виде цветных графических иллюстраций и текстовых комментариев.

Графический интерфейс программы ESID в одном из рабочих режимов показан на рис. 6. Как видно из рисунка, перечень диагностических признаков отображается в левой части экрана монитора, перечень возможных патологий – в правой. Сортировка наименований патологий производится в порядке убывания весовых коэффициентов при наименованиях патологий. Весовые коэффициенты вычисляются с использованием элементов Байесовской стратегии и представляют собой апостериорные вероятности заболеваний для совокупности наблюдаемых признаков.

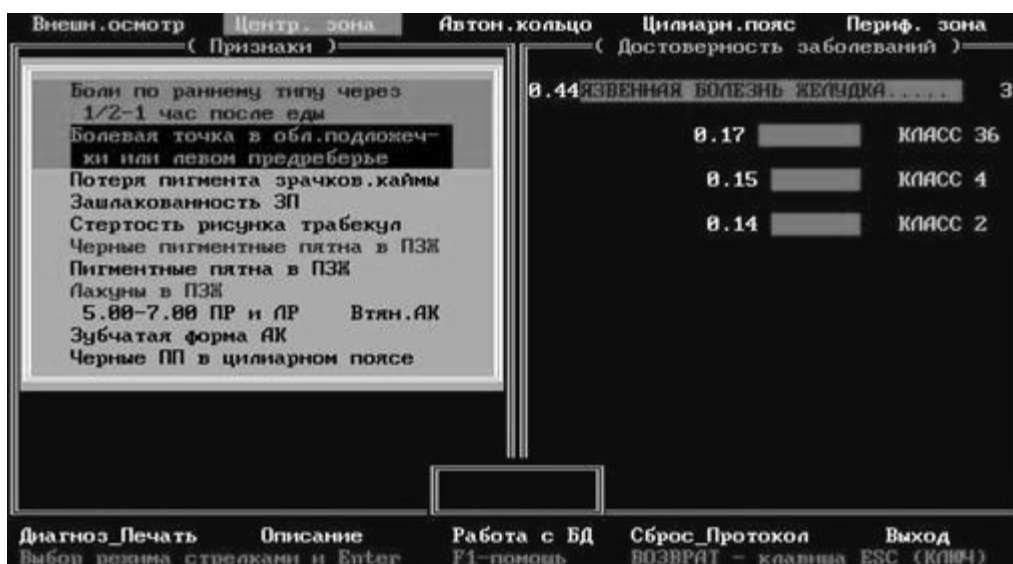


Рис. 6. Графический интерфейс программы ESID.

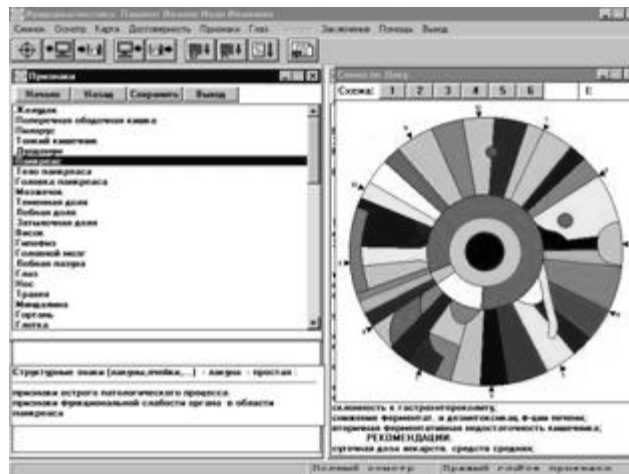
Практика эксплуатации программы ESID в медицинском центре Киевского ГНВП «Аргус» показала, что практикующих иридологов интересует не столько «вероятностная» сортировка наименований патологий, сколько сортировка наименований патологий по принадлежности к той или иной системе (сердечно-сосудистой, дыхательной и т.п.). Вторым недостатком программы ESID – отсутствие графических изображений схем проекционных зон, позволяющих быстро (и в естественной форме) указывать местоположение топических признаков.

Компьютерная программа ESID-2 Киевского ГНВП «Аргус». Компьютерная программа ESID-2 (Windows), разработанная в 1995-96гг. [10], [11], является составной частью аппаратно-программного иридодиагностического комплекса АРМИР и представляет собой экспертную систему, в значительной степени свободной от недостатков программы ESID. Система предназначена, в первую очередь, для практикующих иридологов, хотя может использоваться и в целях обучения, и в научно-исследовательских целях.

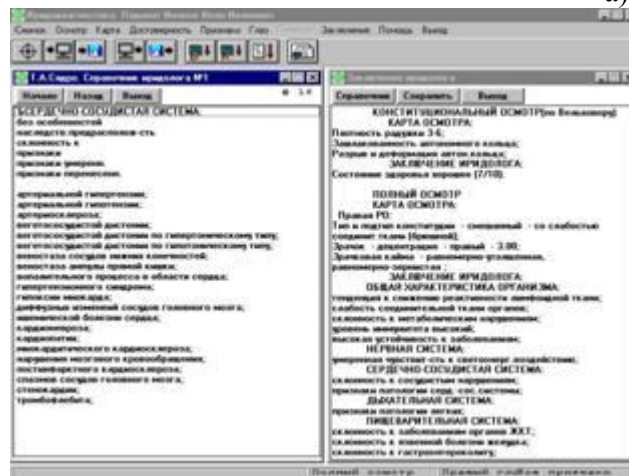
Важной характеристикой системы является ее полезность для иридологов практически любого уровня квалификации. Основные возможности системы:

- ввод в память компьютера цветных изображений радужных оболочек и их отображение на экранах компьютера и вспомогательного монитора в 256 цветовых градациях;
- автоматизированное формирование карты осмотра и заключения иридолога;
- запись результатов обследования в архив на жестком магнитном диске;
- корректировка имеющихся электронных справочников иридолога либо создание новых справочников (модернизация баз данных осуществляется самим пользователем);
- просмотр, редактирование архивных записей;
- печать результатов диагностики.

Графический интерфейс программы ESID-2 в режиме ввода информации о наличии топических признаков и в режиме редактирования заключения показан на рис. 7 (а) и (б) соответственно.



а)



б)

Рис. 7. ESID-2: а) ввод информации; б) редактирование заключения

В программе ESID-2 использован механизм логического суммирования, с последующей сортировкой, наименований патологий.

Практическая эксплуатация программы ESID-2 подтвердила эффективность принятых медико-технических решений. Время формирования заключения иридолога составляло примерно 10 минут при углубленном осмотре и примерно 3-5 минут при скрининг-обследованиях. Объем заключения не превышал 1 листа формата А4, при этом форма заключения была понятной как врачам-клиницистам, так и пациентам.

Компьютерная программа ESID-3 НТУУ (КПИ) и УАНМ. Компьютерная программа ESID-3 разработана в 2000-2001 годах, модернизирована в 2003 году, специалистами НТУУ (КПИ) и УАНМ А.Н.Продеус и Т.А.Сядро. По своему назначению и функциональным возможностям программа ESID-3 близка к программе ESID-2, превосходя ее по продуманности интерфейса, что делает ее значительно более удобной в эксплуатации.

На рис. 8 показан графический интерфейс программы ESID-3 в режиме ввода информации о наблюдаемых признаках.

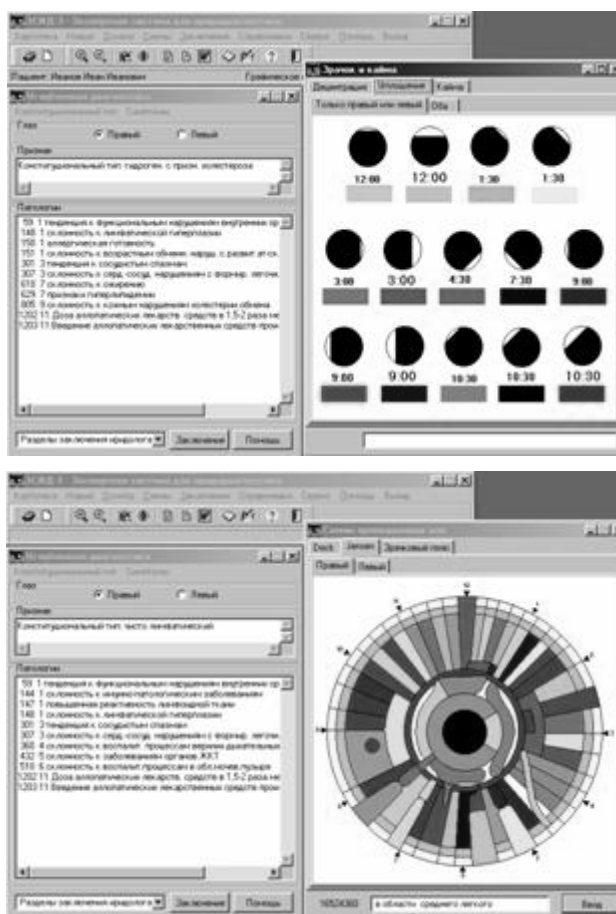


Рис. 8. ESID-3: режим ввода информации

Программа ESID-3 содержит систему встроенных электронных справочников, – иридолога, диетолога, гомеопата, – а также позволяет создать и эксплуатировать неограниченное количество собственных электронных справочников пользователя. Данную программу целесообразно использовать как ядро программного обеспечения современного АРМ'а иридолога. Именно в таком

качестве она используется, например, в АРМ'е иридолога Киевского Финансового лица [12], [13]. Имеется локализация программы на английском языке.

Компьютерная программа для иридодиагностики Михаила и Георгия Абрамовых. Абрамов Михаил Самуилович – автор изданной в 1991 в Ташкенте в изд. «Ибн Сина» книги «Иридодиагностика с компьютером» [14]. В настоящее время проживает в США, сотрудничает с канадским нейро-оптическим институтом Брайена Марчиа.

Компьютерная программа для иридодиагностики «SoftIris Pro» – англоязычная программа для иридодиагностики на IBM-совместимом ПК. Ее графический интерфейс показан на рис. 9. Программа позволяет осуществлять наложение изображений радужной оболочки и схемы проекционных зон. Левая часть окна предназначена для работы с текстовой информацией (ввод данных о пациенте, ввод информации о наблюдаемых признаках, сформированное заключение), правая – с графической (изображений радужки, схемы проекционных зон или сумма этих изображений).

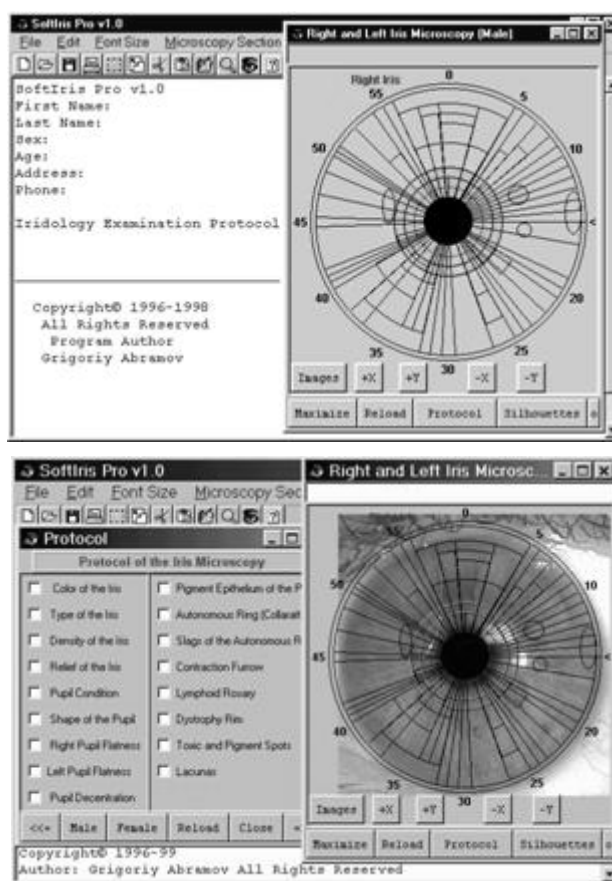


Рис. 9. Графический интерфейс программы «SoftIris Pro»

Графический редактор для иридологов GIDRA НТУУ (КПИ). Разработанная К.Н. Коваленко и А.Н. Продеусом в 2002-2004 годах на кафедре акустики и акустоэлектроники НТУУ (КПИ), компьютерная программа GIDRA (рис. 10) представляет собой графический редактор, учитывающий требования иридологов различных категорий – практических иридологов, иридологов-исследователей и иридологов-преподавателей.

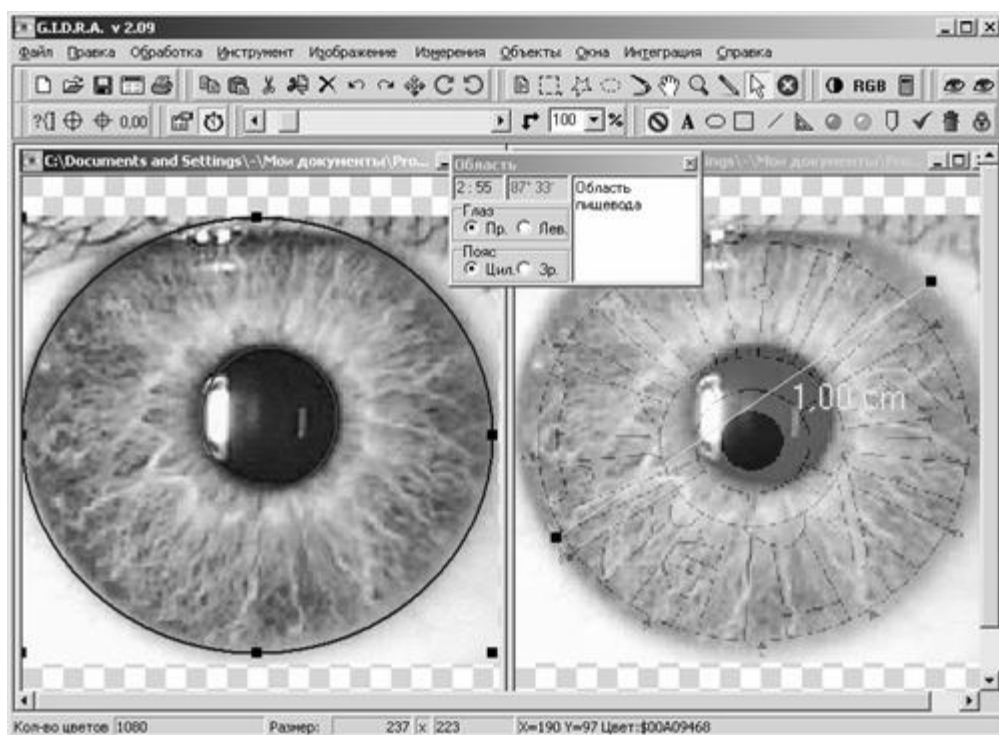


Рис. 10. Графический интерфейс программы GIDRA

Программа позволяет совмещать изображения схемы проекционных зон с изображениями радужной оболочки. Можно измерять координаты наблюдаемых признаков в привычной для иридологов системе координат «часовой циферблат», а также измерять длину, периметр и площадь иридологических знаков. Пользователь может делать надписи на изображениях, рисовать на них прямоугольники, овалы, линии и стрелки, выделять, вырезать или обрезать выделенный фрагмент изображения и т.п. Удачно решена функция быстрого масштабирования с помощью элемента управления в виде ползунка. Возможны быстрые повороты изображения на 90 градусов, а также плавное вращение изображения. Возможна фильтрация выделенных участков изображения, позволяющая подчеркнуть структурные особенности изображения. Имеется функция автоматизированного оконтуривания знаков радужки (режим «волшебная палочка»).

Программы «IriSearch Pro» и «IriSearch Plus» (Австралия: Roodu’Jardin, Naturopathic Technologies). Это программный инструмент для практикующих иридологов, а также для иридологов-исследователей и иридологов-педагогов. Содержит функции обработки изображений, захватываемых с помощью цифрового иридоскопа фирмы Naturopathic Technologies, либо поступаемых от иных источников – цифровых фотокамер и сканеров. Иридологам-преподавателям предназначены такие функции обработки изображения как создание надписей, указателей, окантовок. Имеются важные для проведения измерений функции: калибровка монитора, контроль цветового баланса, наложение схем проекционных зон, вращение изображений, расчет среднего цвета радужки, измерительный инструмент, каталог изображений, формирование отчетов, выдача рекомендаций.

Программа IrisCamera фирмы Miles Research. Американская фирма Miles Research разрабатывает оборудование и программное обеспечение для офтальмологии и оптометрии. На рис. 11 показан графический интерфейс компьютерной программы IrisCamera, обеспечивающей захват и сохранение в памяти компьютера изображений с выхода видеокамеры JVC (2/3" RGB видеокамера с композитным выходом и, опционально, выходом S-video). Для перекачки информации из видеокамеры в компьютер используется кабель с разъемами DB-9.



Рис. 11. Графический интерфейс компьютерной программы IrisCamera

Анализ данного ПО свидетельствует о принципиальной возможности получения качественного изображения с помощью видеокамеры при условии, что используемое оборудование позволяет осуществить тщательную регулировку баланса белого цвета. На рис. 12 приведены примеры несбалансированного (а) и сбалансированного (б) изображений.

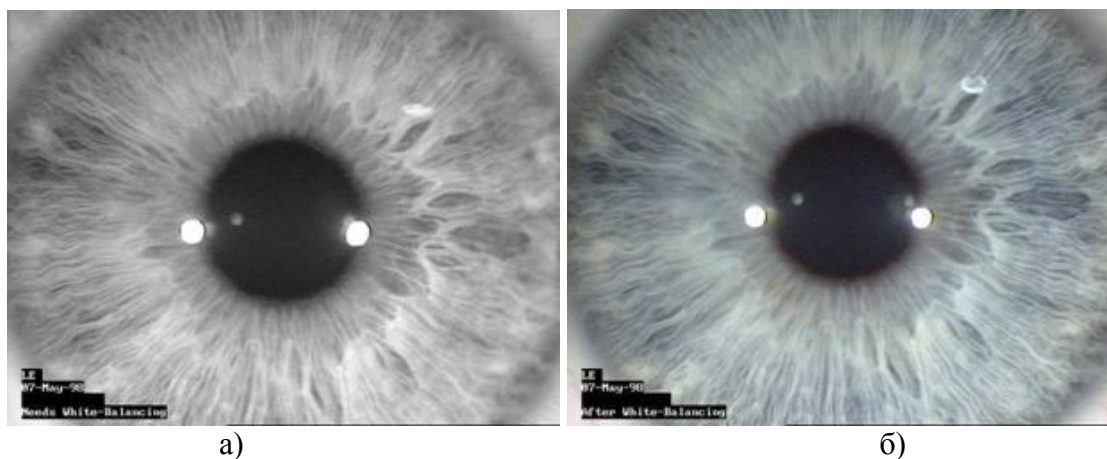


Рис. 12. Несбалансированное (а) и сбалансированное (б) изображения

Компьютерная программа канадского нейро-оптического института (CNRI). Фрагменты графического интерфейса программы канадского нейро-оптического института (CNRI) показаны на рис. 13.

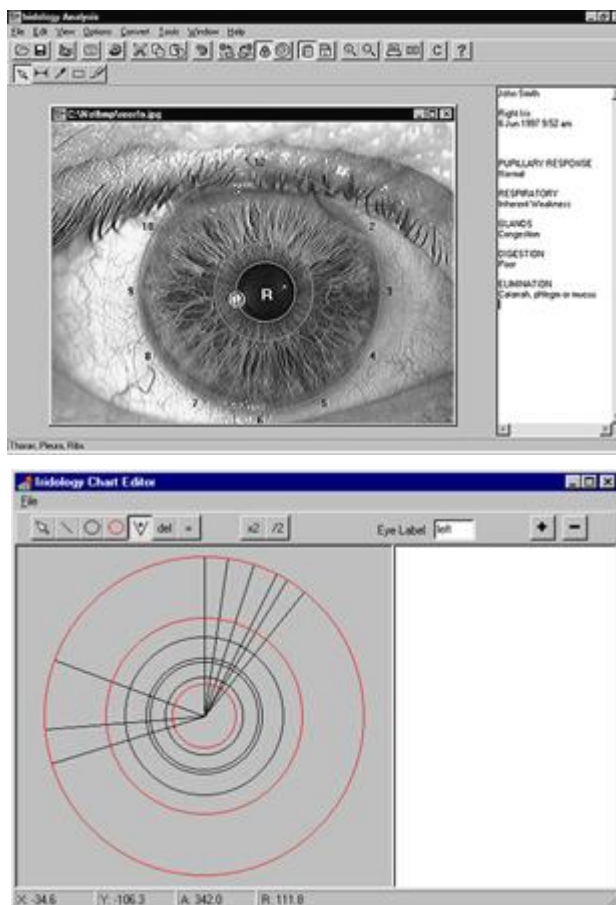


Рис. 13. Графический интерфейс программы нейро-оптического института (CNRI)

Данное программное обеспечение обладает довольно скромными функциональными возможностями.

2.8.2. Программа-лаборатория TILDA для научных исследований

Принадлежащие двум американским исследователям Матту Карвовски (Matt Karwowski) и Дану Ванеку (Dan Waniek) веб-сайты Iris-Works(TM) и Iris-Ward(TM) в 2002 году объявили о проведении исследовательской программы под названием «Предварительное изучение микроизменений в моделях радужек» («Preliminary Study of Micro-changes in Iris Patterns» (PSoM-CIP)). Авторы данного проекта к «микроизменениям» относят любые индивидуальные изменения ириса, случившиеся в процессе заболевания. Как полагают авторы проекта, эти изменения тяжело обнаружить или измерить без применения специальных процедур цифровой обработки изображений, подавляющих влияние маскирующих факторов, таких как: различные условия освещения, хроматические характеристики фотопленки, размер и положение зрачка.

В результате «увязывания» разных научных программ обработки изображений и популярных графических редакторов в пакет, управляемый из

единого меню, была получена компьютерная «программа-лаборатория» обработки изображений, названная TILDA.

Безусловный интерес представляет попытка Матта Карвовски и Дана Ванека применить к изображению радужной оболочки преобразование Фурье, затем подавить «вредные» частотные составляющие и, после такой фильтрации изображения, вернуться в область изображения (рис. 14). Комментируя данные изображения, Матт Карвовски и Дан Ванек пишут, что подобный феномен можно использовать при распознавании некоторых особенностей радужки. Приведенный пример уникален в том смысле, что до настоящего времени исследователи почти не интересовались вопросами применения преобразования Фурье к изображениям радужки.

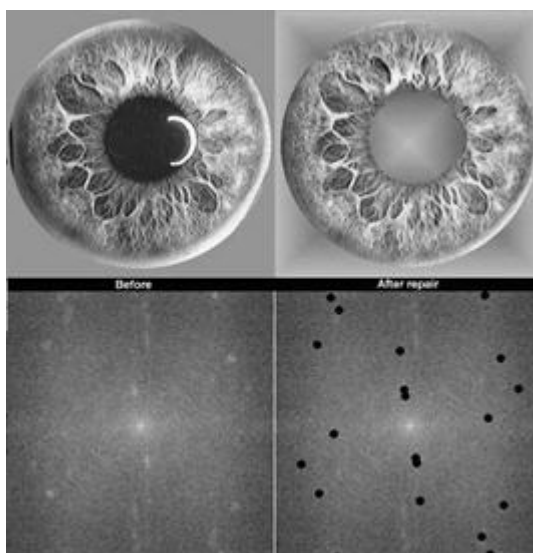


Рис. 14. Цифровая фильтрация с помощью двумерного преобразования Фурье

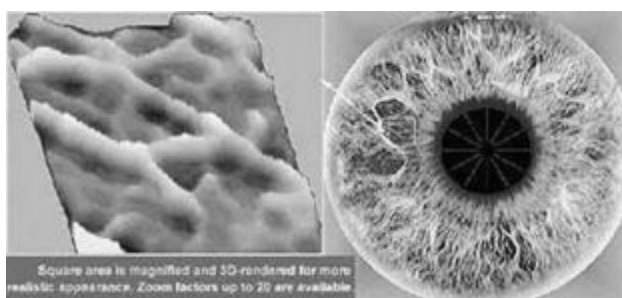


Рис. 15. Цифровая обработка изображения: псевдо-трехмерный рельеф

Не меньший интерес представляют: создание псевдо-трехмерных фрагментов изображения радужки (рис. 15), получение ярких сечений радужки (рис. 16 (а) и (б)), псевдо-расцвечивание радужки (рис. 16 (в)), двумерная фильтрация изображений (рис. 16 (г)).

2.8.3. Рабочие станции и специальное оборудование

Рабочая станция – это совокупность аппаратного и программного обеспечения. В отечественной литературе вместо термина «рабочая станция» предпочитают использовать термин «автоматизированное рабочее место» (АРМ).

АРМ'ы иридолога могут быть выполнены на базе настольных ПК либо на базе переносных компьютеров (Notebook, Laptop). При этом для ввода изображений радужки в память компьютера используются видеокамеры либо цифровые фотокамеры. Разумеется, для использования видеокамеры необходима специальная и, следовательно, более дорогая видеокарта компьютера, содержащая аналого-цифровой преобразователь (АЦП) сигнала с выхода видеокамеры. Поскольку цифровая фотокамера содержит собственный АЦП, требования к видеокарте компьютера снижаются. Примеры рабочих станций: 1) АИК-02 (Россия: Москва, МПГУ, МВНТП «ЛЭМО»); 2) «СИГМА-ИРИС» (Россия: Москва, инновационное научно-производственное предприятие «СИГМА-А»); 3) «АРМИР» (Украина: Киев, медцентр «ЛІДІР»); 4) «NT Digital Iriscope Pro» и «NT Digital Iriscope Notebook Pro» (Австралия: Roo du'Jardin, Naturopathic Technologies); 5) «Bexel IRINA» (Канада: Canadian Neuro-Optic Research Institute) [8].

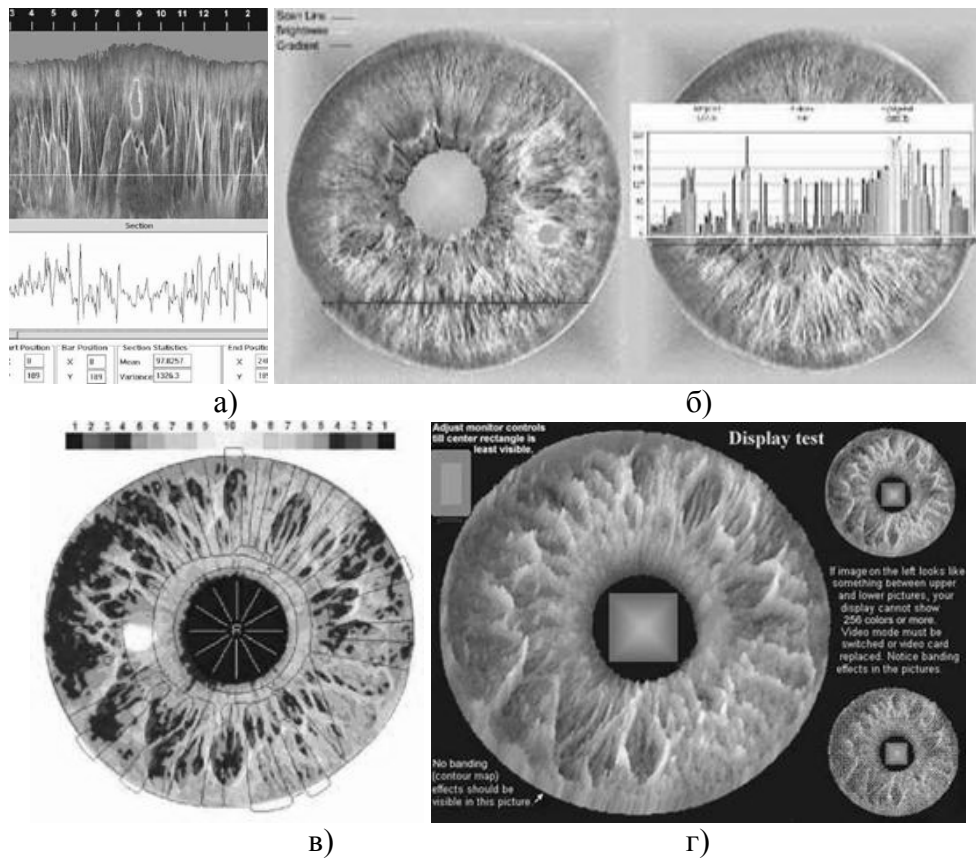


Рис. 16. Разные варианты обработки изображения:

- а) и б) яркостное сечение радужки;
- в) изображение радужки в псевдоцвете;
- г) фильтрация изображений радужки

Примеры применяемых видеокамер: 1) «IVC-1» (США: Miles Research, Instituto Prana); 2) «OSCAR» (Украина: Киев, медцентр «ЛІДІР») [8].

Примеры применяемых цифровых фотокамер: 1) «IDC-1 Fuji Digital Iris Camera» (США: Miles Research, Instituto Prana); 2) «MVC-FD91» (Канада: Canadian Neuro-Optic Research Institute); 3) «Nikon CoolPix 950» (Канада: Canadian

Neuro-Optic Research Institute); 4) «DV2000E» (США: San Francisco, Joyful Living Services). В приведенном перечне соседствуют как специально приспособленные для иридодиагностики профессиональные цифровые фотокамеры (например, «IDC-1 Fuji Digital Iris Camera»), так и бытовые цифровые фотокамеры (например, «MVC-FD91») [8]. Отметим, что стоимость профессиональных камер существенно выше стоимости бытовых – в несколько, а порой и в десятки раз.

Ниже рассмотрены характеристики некоторых АРМ'ов и применяемого в них специального оборудования.

АРМ иридолога Е.С. Вельхова. В 1985г. в лаборатории проф. Е.С. Вельхова Университета дружбы народов им. П. Лумумбы разработана первая в СССР автоматизированная система для иридодиагностических исследований [1]. В состав системы вошли видеокамера, аппаратно-программные средства ввода видеосигнала в компьютер, компьютер типа «Электроника 60 М» со специальным программным обеспечением, вспомогательный монитор.

Функции программного обеспечения этой системы: предварительная обработка изображения радужки, определение азимутальных зависимостей размеров зрачка и радужки, оценка информативных признаков (цвет, пространственная структура локальных знаков, оценка статистических характеристик иридологических признаков, определение координат иридологических признаков и их отождествление со знаками иридологических схем).

Комплекс «ИРИС». В конце 80-х, начале 90-х годов в той же лаборатории, в содружестве с Московским НПО «Комета», разработан комплекс «ИРИС» на базе IBM-совместимых компьютеров [8], [1]. Назначение комплекса – проведение научных исследований и практическая иридодиагностика в стационарах и крупных диагностических центрах. Наличием исследовательской цели поясняется развитость части программного обеспечения, направленной на обработку изображений радужки: фильтрация изображений, масштабирование фрагментов, поворот изображения, оконтуривание зрачка и автономного кольца, псевдорасцвечивание полутонового изображения и др. Для практической иридодиагностики предназначены режимы топической диагностики, системного анализа, оценки степени риска развития определенной патологии (печени, поджелудочной железы, желудка и др.), оценки рейтинга старения и наследственного статуса человека.



Рис. 17. Внешний вид комплекса «ИРИС».

На рис. 17 показан внешний вид комплекса «ИРИС», а на рис. 18 показаны результаты цифровой обработки изображений радужки в этом комплексе: фильтрация прямоугольного фрагмента изображения радужки (рис. 18 (а)), автоматическое оконтуривание автономного кольца (рис. 18 (б)), изменение масштаба выделенного фрагмента изображения (рис. 18 (в)).

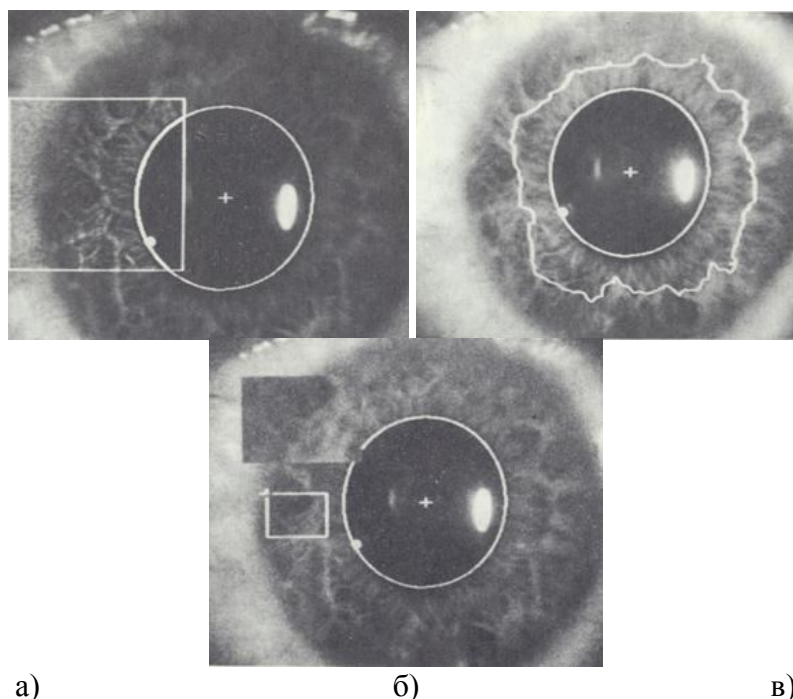


Рис. 18. Фильтрация (а), оконтуривание (б), а также изменение масштаба фрагмента изображения (в)

Характерным примером частной диагностики, производимой на комплексе «ИРИС», является программа раннего выявления рака желудка, построенная на анализе 50 признаков: 12 клинических, 6 анамнестических, 21 иридологических, 11 физиогномических (как видим, собственно иридологических признаков в данном случае меньше половины). Производится интегральная оценка риска по раку желудка: легкая степень 5-30 баллов, средняя степень 31-60, значительная степень – 61 и выше. На основании клинических и иридо-топографических симптомов предположительно указывается топика опухолевого процесса (кардия, тело, антрум желудка).

Последовательность действий врача-иридолога выглядит следующим образом:

- выбор схемы проекционных зон органов (схемы Е.С. Вельхова, В. Jensen, К. Gunter, J. Angerer, R. Bourdiol, J. Deck;
- наложение схемы проекционных зон на изображение радужки;
- анализ фрагментов изображения.

Распечатываемые результаты диагностики состоят из двух частей: графического изображения радужки и текстового заключения. Время диагностики на «Ирисе» составляет 5-30 мин.

В [14] отмечены такие недостатки комплекса «ИРИС» как сложность и громоздкость, препятствующие его применению даже в поликлиниках и больницах, не говоря уже о применении в мобильном варианте, например, врачами «Скорой помощи», для экспресс-диагностики. Небезупречна и форма выдаваемого заключения [1]: карта осмотра имеет вид последовательности кодов типа «2а», «2б» и т.д., что осложняет расшифровку заключения другими специалистами.

Автоматизированная система «ИРА». Менее громоздким аналогом комплекса «ИРИС» является автоматизированная система «ИРА» (разработка той же лаборатории). Однако в [14] отмечается, что «ИРА» все же менее доступна лечебным учреждениям, чем иридография (анализ увеличенных изображений радужки, зафиксированных на цветных слайдах), по причине высокой стоимости комплекса.

АРМ врача-иридолога. С 1993г. осуществляется коммерческое распространение иридодиагностического комплекса «АРМ врача-иридолога», разработанного в научно-производственной фирме «БЛИЦ» и АО «РИНК» при участии Е.С. Вельхова. В состав этого АРМ'а входят: бинокулярный иридокоп, цветная TV камера с переходным объективом (позволяющим объединить TV камеру с иридокопом), осветитель волоконный, персональный IBM-совместимый компьютер, устройство ввода и обработки сигналов изображений (устанавливается в системный блок ПК) и соответствующее программное обеспечение. Назначение АРМ'а – автоматизация практической иридодиагностики, обучение и переподготовка иридодиагностов.

Программное обеспечение комплекса позволяет осуществлять автоматизированное выявление патологий (частная диагностика) по следующим системам организма: сердечно-сосудистая, поджелудочная железа, легкие, печень, почки, желудочно-кишечный тракт, молочная железа, предстательная железа, портальная гипертензия. В целом комплекс обеспечивает запись, хранение, обработку и анализ изображений радужной оболочки глаза, архивирование и печать заключений, статистических и регистрационных данных осмотра пациентов.

Автоматический иридодиагностический комплекс АИК-02. Разработанный в МВНТП «ЛЭМО» (Россия: Москва) АИК-02 предназначен для проведения массового медицинского обследования населения. Комплекс АИК-02 может поставляться в мобильном исполнении для осуществления осмотров в удаленных и труднодоступных регионах. В состав комплекса входит: персональный IBM-совместимый компьютер; цветная видеокамера; телевизор; оптико-визуальный комплекс; программное обеспечение (программа «Топика»). При функционировании комплекса в автоматическом режиме, на изображении радужки определяются границы зрачка, зрачковой каймы, автономного кольца, края радужки, в результате чего определяются: абсолютные и относительные размеры зрачка, зрачковой каймы, зрачкового и цилиарного поясов на изображении ириса; децентрация зрачка, зрачковые уплощения, зрачковые деформации (эллипсовидность); средняя ширина зрачковой каймы, определение типа ее формы; средняя ширина автономного кольца, определение мест разрывов, втяжений и выбуханий. Для повышения достоверности принимаемых решений имеется диалоговый режим, в котором оператор может откорректировать результаты автоматически формируемых решений.

АРМ иридолога «СИГМА-ИРИС». Анонсируется как автоматизированное рабочее место иридолога-эндоэколога (Россия: Москва, инновационное научно-производственное предприятие «СИГМА-А»), обеспечивающее: скрининг-обследование общего состояния организма; неинвазивный анализ общего состояния организма; коррекцию состояния организма.

Состав системы «СИГМА-ИРИС»:

- персональный компьютер в стандартной комплектации;
- столик изменяемой высоты;
- видеокамера и подсветка;
- фреймграббер для оцифровки изображения;
- автоматика и электроника управления, контроллер;
- модем для обмена информацией;
- лазер для квантовой иридоотерапии.

Внешний вид АРМ'а показан на рис. 19.

Функции программного обеспечения системы «СИГМА-ИРИС»:

- ведение электронной картотеки и архивов;
- захват изображений радужной оболочки;
- анализ изображений радужной оболочки;
- частные диагнозы и анализ эндоэкологического состояния;
- квантовая иридоотерапия.

О весьма низком качестве получаемых изображений можно судить по рис. 20 – очевидно, нарушен цветовой баланс, в результате на рис. 20 (а) преобладает красный цвет, а рис. 20 (б) вообще представляется практически черно-белым. Пространственное разрешение изображений позволяет судить лишь о структурных знаках достаточно больших размеров.

На рис. 21 (а) демонстрируется возможность отображения на мониторе схемы проекционных зон, а на рис. 21 (б) – вид комплекса «СИГМА-ИРИС» в режиме лазерной (квантовой) иридоотерапии.



Рис. 19. Внешний вид АРМ'а «СИГМА-ИРИС»

АРМ иридолога М.Абрамова. В [14] описывается АРМ врача-иридолога, в состав которого входят: щелевая лампа типа ЩЛ-2Б персональный компьютер типа «Атари-65ХЕ» либо IBM-совместимый, специальное программное обеспечение.

Методика обследования с использованием такого АРМ'а:

- в компьютер вводится, в диалоговом режиме работы, информация о тополабильных иридопризнаках – цвет, плотность, тип радужки, размеры и деформация зрачка и т.п.;
- выбирается оптимальная схема проекционных зон и производится ввод в компьютер информации о топических иридопризнаках;
- автоматически формируется врачебное заключение (по результатам ввода в компьютер информации о наблюдаемых иридопризнаках);
- осуществляется редактирование сформированного текста врачебного заключения, который затем протоколируется на магнитном носителе либо распечатывается на бумаге.

В [14] констатируется, что работа с АРМ'ом способствует возникновению у врача четкой системы мышления, базирующейся на избранном диагностическом алгоритме. При этом время обследования одного пациента сокращается почти в 2 раза, а точность диагностики существенно повышается. Очевидно, к настоящему времени данный комплекс морально устарел.

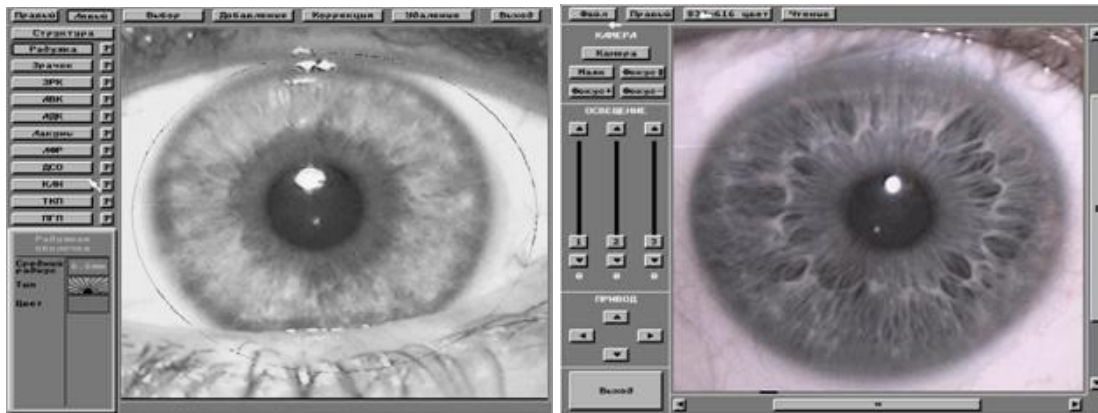


Рис. 20. «СИГМА-ИРИС»: а) изображение №1; б) изображение №2

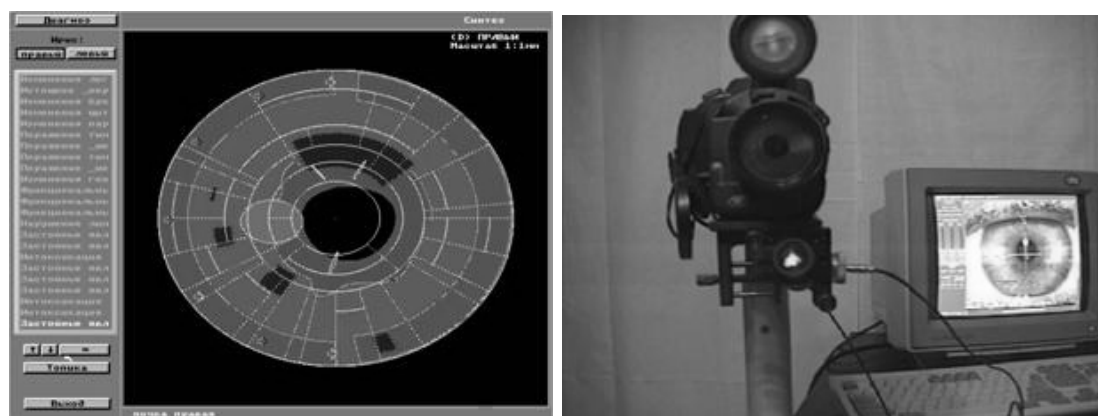


Рис. 21. «СИГМА-ИРИС»: а) изображение схемы; б) иридотерепия

АРМ иридолога О.В. Петенко и Н.И. Гречишниковой. В [6] описывается система компьютерной иридодиагностики, реализующая ввод

видеосигнала с выхода телевизионной камеры в IBM-совместимый компьютер и позволяющая автоматизировать формирование заключения иридолога.

По мнению авторов [6], сопряженные с компьютером современные телевизионные системы не позволяют обеспечить требуемую разрешающую способность анализа изображения радужки на мониторе компьютера – виной тому существующие стандарты формирования телевизионного изображения (625 строк на кадр), не дающие возможности надежно идентифицировать мелкие образования радужки (пигментные пятна, лакуны и т.п.).

При описании возможностей программного обеспечения в [6] подробно охарактеризован лишь принцип автоматизированного формирования заключения иридолога – по-видимому, этот режим является наиболее развитым по сравнению с остальными режимами функционирования компьютерной программы. Принцип этот достаточно прост: для каждой системы организма человека (нервная, сердечно-сосудистая, бронхо-легочная и т.д.) на мониторе отображаются тексты вариантов стандартных предложений, среди которых врач осуществляет выбор нужного варианта. При этом, одновременно (автоматически) с текстом выбранного предложения, в заключение переносится и текст пояснений использованных терминов. Кроме того, автоматически в текст заключения переносятся рекомендации по дополнительным обследованиям, по лечению, питанию и профилактике.

Объем сформированного таким образом заключения может достигать 10 страниц машинописного текста – главным образом из-за наличия пояснительного текста, адресуемого пациенту (объем пояснений достигает 85% общего объема заключения). Чрезмерная объемность заключения, по-видимому, является существенным, хотя и легко устранимым, недостатком рассмотренной системы.

Комплексы «Ирис» и «Ирис-ТМ» Киевского НПО «Квант». В конце 80-х годов Киевское НПО «Квант» являлось одним из крупнейших государственных разработчиков технических средств иридодиагностики. Поэтому представляет интерес рассмотрение характеристик двух разработок этого предприятия – комплексов «Ирис» и «Ирис-ТМ».

Комплекс аппаратуры для иридодиагностики «Ирис» разработан в 1989 году с участием В.В. Кривенко – руководителя специализированной лаборатории нетрадиционных методов диагностики и терапии Института проблем онкологии им. Р.Е. Кавецкого АН Украины [2].

Конструктивно комплекс выполнен в виде трех отдельных приборов: преобразования и передачи изображения, предварительной обработки и хранения изображений, окончательной обработки изображения.

Схема функционирования комплекса вкратце такова. После отображения на экране монитора подсвеченной радужки, ее изображение совмещается с полупрозрачным, заранее выбранным, трафаретом ириса (схемой проекционных зон), механически закрепляемом на экране монитора. Совмещение достигается с помощью подвижной платформы устройства юстировки, перемещающейся вместе с передающей телекамерой в трех пространственных координатах. В момент совмещения изображение запоминается видеоманитофоном, работающим в режиме «стоп-кадра».

В приборе предварительной обработки и хранения изображений реализуются возможности создания электронного трафарета и трансформации изображения радужки (изменение масштаба изображения, выделение фрагментов

изображения, наложение на изображение графической и текстовой информации в виде графиков, гистограмм, числовых трафаретов, данных о пациенте, даты обследования, заключения врача-иридолога).

Прибор обработки изображения радужки содержит персональный компьютер (ЕС-1842), устройство ввода графических изображений (СМП 6410), накопитель на магнитной ленте (СМ 5300.01). Назначение этого прибора – разработка и отладка программного обеспечения для полуавтоматической и автоматической постановки диагноза.

Функции программного обеспечения: определение координат иридо-знаков, оценивание степени децентрализации и уплощения зрачков, построение гистограмм по цвету или яркости, смешивание изображений, нахождение центра плотности в выбранном фрагменте изображения, корректировка изображения, статистическая обработка изображений, вводимых со слайдов.

В настоящее время, когда возможности аппаратного и программного обеспечения вычислительной техники существенно возросли, становятся очевидными моральная устарелость и громоздкость комплекса «Ирис».

Следующей и более современной разработкой НПО «Квант» явился комплекс «Ирис-ТМ», предназначенный для проведения массовых профилактических обследований пациентов и для обучения врачей.

Состав «Ирис-ТМ»: цветная телевизионная камера с оптическим устройством для макросъемки, электронный блок для ввода видеосигнала в компьютер, IBM-совместимый компьютер, видеомэгнитофон, специальное программное обеспечение.

Основные возможности программного обеспечения: управление вводом видеосигнала в компьютер, выбор зоны интереса (окна), работа с зоной интересов (масштабирование), сохранение изображения на ГМД и жестком диске, ввод изображения с ГМД или жесткого диска, анализ изображения (выделение контуров, работа с гистограммой яркости, определение геометрических характеристик изображения), наложение трафаретов иридологических карт на изображение радужной оболочки глаза пациента, получение справочной иридо-диагностической информации.

По данным разработчиков «Ирис-ТМ», качество изображения радужки существенно зависит от типа видеокамеры. Недорогие бытовые видеокамеры не удовлетворяют требованиям иридологов, а профессиональные видеокамеры весьма дороги. Вывод разработчиков – разумным компромиссом между ценой и качеством является использование импортной бытовой телевизионной техники.

Производство комплексов «Ирис-ТМ» прекращено в начале 90-х годов.

АРМ иридолога «АРМИР» Киевского ГНВП «Аргус». Программно-аппаратный комплекс «АРМИР» разработан в 1995-96гг. в Киевском ГНВП «Аргус» [10], [11] для Киевского медицинского диагностического центра «ЛІДІР». Внешний вид комплекса представлен на рис. 22. Комплекс «АРМИР» эксплуатируется как автономно, так и в локальной сети. При функционировании в локальной сети возможен многосторонний обмен информацией между четырьмя АРМ'ами: иридолога, УЗИ-специалиста, кардиолога, терапевта.

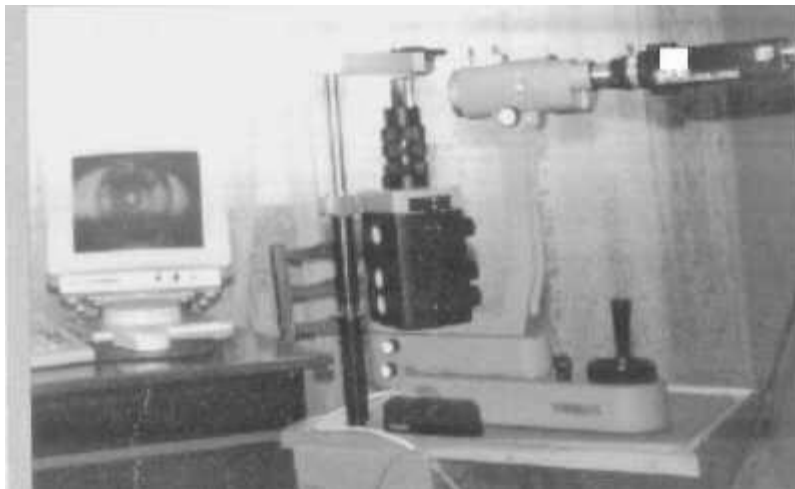


Рис. 22. Комплекс «АРМИР»

В состав оборудования комплекса «АРМИР», помимо ПК, входит щелевая лампа и сопряженная с ней аналоговая видеокамера «OSCAR». Поскольку на момент создания АРМ'а в продаже отсутствовали недорогие видеокарты для ввода изображения в компьютер, для этой цели по специальному заказу на одном из киевских предприятий было разработана аппаратно-программная система оцифровки и ввода изображений в компьютер.

Принципиальным недостатком данной системы было фактическое отсутствие режима цветовой калибровки – так называемый баланс белого цвета оценивался «на глаз» и достигался путем ручной регулировки параметров системы. Другим существенным недостатком этой системы была нестабильность работы – по мере разогрева аппаратуры баланс белого цвета нарушался.

Пример изображения, получаемого такой системой, приведен на рис. 23. Очевидно нарушение цветового баланса (преобладает зеленый цвет) и низкая разрешающая способность (волокнистая структура радужки практически не видна).

По результатам эксплуатации комплекса «АРМИР» были сделаны следующие выводы:

- использование недорогих бытовых видеокамер не позволяет добиться хорошего качества изображения;
- оцифровку видеосигнала следует производить с помощью качественных видеокарт.

Рабочие станции Гари Озарко (Австралия). На персональном веб-сайте Гари Озарко (Gary Ozarko) сообщается об аппаратно-программных иридологических системах, в состав которых входят: цифровая фотокамера с насадочными линзами для иридологической макросъемки, световолоконная система освещения, подставка для головы пациента, компьютер и специальное программное обеспечение. Опционально (т.е. по желанию) могут быть приобретены: цифровые видеокамеры, цифровая 35мм фотокамера, щелевая лампа или иридоскоп. Вся аппаратура снабжена насадочными линзами для иридологической макросъемки.

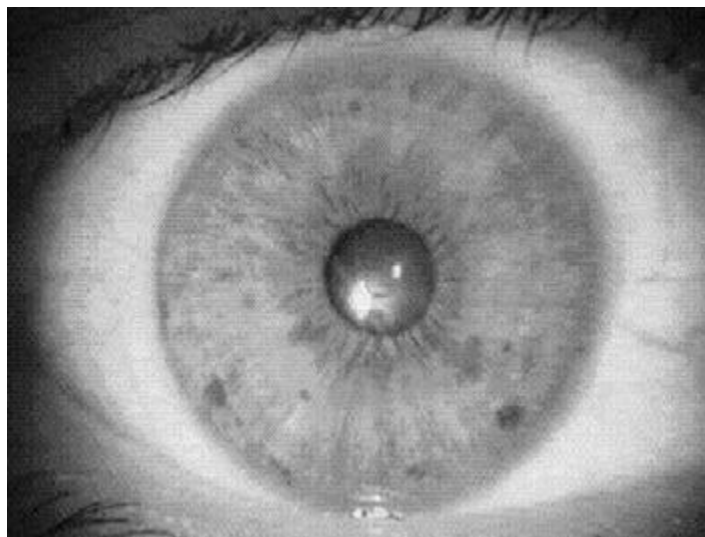


Рис. 23. Изображение радужки в комплексе «АРМИР»

Рабочие станции фирмы Naturopathic Technologies, Roo du'Jardin (Австралия). Рабочая станция «NT Digital Iriscope Notebook Pro» – это ноутбук, соединенный с видеокамерой, установленной на специальной платформе (рис. 24 (а)). Рабочая станция «NT Digital Iriscope Pro» (рис. 24 (б)) – разновидность системы, использующей настольный IBM-совместимый компьютер.

Рабочие станции Брайена Марчия (Канада). В канадском нейро-оптическом институте Брайена Марчия (Bryan K. Marcia) разработаны:

- аппаратно-программный комплекс для иридодиагностики (рис. 25 (а)): компьютер, цифровая фотокамера или видеокамера, подставка для головы пациента, программное обеспечение;
- аппаратно-программный комплекс для «замораживания» изображений (рис. 25 (б)): цифровая фотокамера или видеокамера, подставка для головы пациента, видеоманитофон;
- аппаратно-программный комплекс «Vexel IRINA» (рис. 26), имеющий вид некоего «сплава» компьютера и электронно-оптической системы.

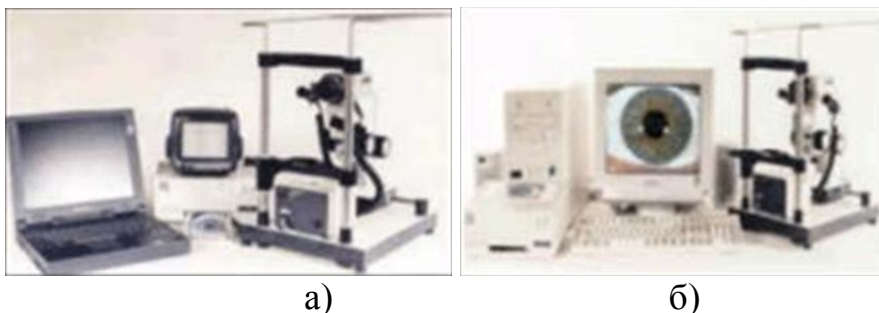


Рис. 24. «NT Digital Iriscope Notebook Pro» (а) и «NT Digital Iriscope Pro» (б)

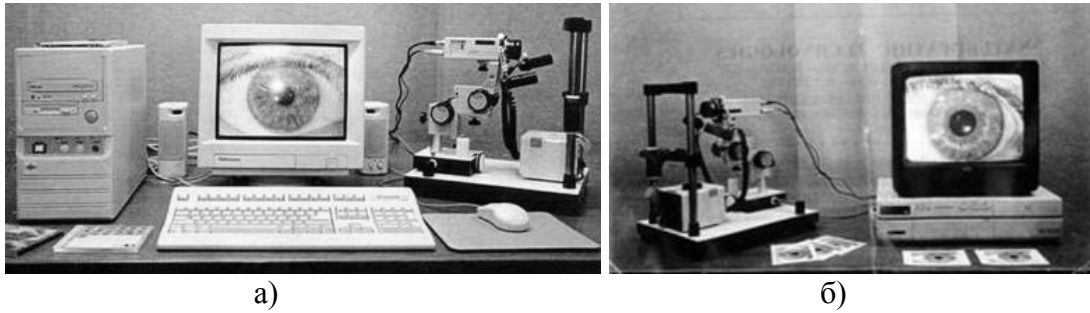


Рис. 25. Рабочая станция (а) и аппаратно-программный комплекс для «замораживания» изображений (б) канадского нейро-оптического института (CNRI)



Рис. 26. Рабочая станция «Bexel IRINA» (CNRI)

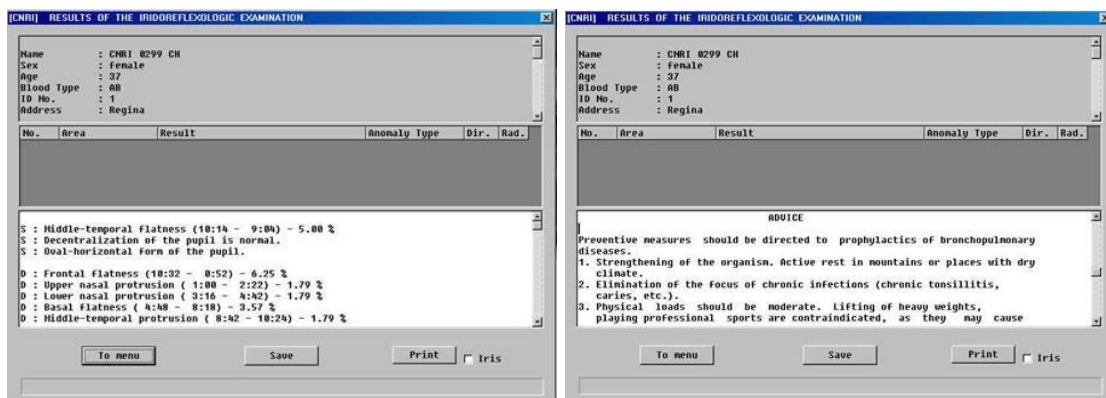


Рис. 27. Рабочий интерфейс «Bexel IRINA»

О некоторых возможностях комплекса «Bexel IRINA» можно судить по рис. 27, где показан графический интерфейс в различных режимах эксплуатации.

АРМ школьного иридолога (Украина). С 1997 по 2002 годы в Киевском Финансовом лицее эксплуатировались компьютерные программы ESID-2 и ESID-3. С 2002 года, после приобретения компьютерной программы-фоторедактора GIDRA, электронных атласов №1 и №2, а также цифровой фотокамеры Casio QV-

8000 SX, стало возможным говорить о создании в Финансовом лицее собственного АРМ'а иридолога [12], [13].

2.8.4. Электронные учебно-методические пособия

Электронные атласы А.Продеуса и Т.Сядро (Украина). В 1999 году выпущено первое издание, а в 2002 году – второе, расширенное и дополненное, издание первого украинского CD-атласа для иридологов. «Электронный атлас иридолога. Часть 1. Иридогенетические синдромы» реализован в виде компьютерной программы с эффективной системой навигации по информационным разделам электронного справочника. Совместное использование атласа и программы ESID-3 позволяет использовать атлас не просто как справочник, но и как средство самотестирования начинающего иридолога. Важным обстоятельством является открытость базы данных атласа для пользователя, который может ее редактировать по собственному усмотрению. На CD, помимо атласа, расположены также: мультимедийные уроки (озвученные видеоролики) в помощь осваивающим атлас и компьютерную программу ESID-3; обзор в html-формате «Компьютерная иридодиагностика – современное состояние»; электронная версия книги А.Продеуса, Е.Захрбовой «Экспертные системы в медицине»; оригинальная компьютерная программа «Электронная картотека иридолога».

Разработанный в 2004 году справочник «Электронный атлас иридолога. Часть 2. Иридологические симптомы» – это естественное продолжение первого справочника, «Электронный атлас иридолога. Часть 1. Иридогенетические синдромы». Особенностью нового атласа является возможность озвучивания выбранного фрагмента текста, позволяющая пользователю, одновременно с анализом изображений радужек, слушать речевой комментарий. Это удобно, поскольку позволяет воссоздать атмосферу «обучения с учителем». Авторы электронных атласов – А.Н.Продеус, Т.А.Сядро.

Графические интерфейсы обоих «Электронных атласов» показаны на рис. 28.

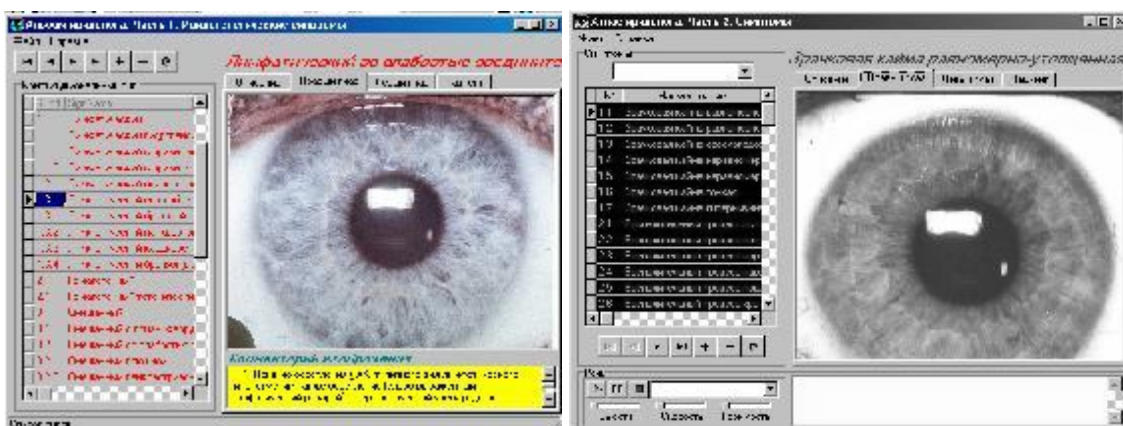


Рис. 28. Графические интерфейсы украинских иридологических электронных атласов

Электронные учебники по иридологии М.Абрамова (США, Россия). Вид раннего издания (1997г.) электронной книги на английском языке «Moder Iridology» М.Абрамова показан на рис. 29 (а).

В 2003г. учебник вышел во второй редакции, а в 2005г. в России анонсирован выпуск новой редакции учебника «Иридодиагностика с компьютером» на русском языке – на рис. 29 (б) показан графический интерфейс этого электронного учебника.

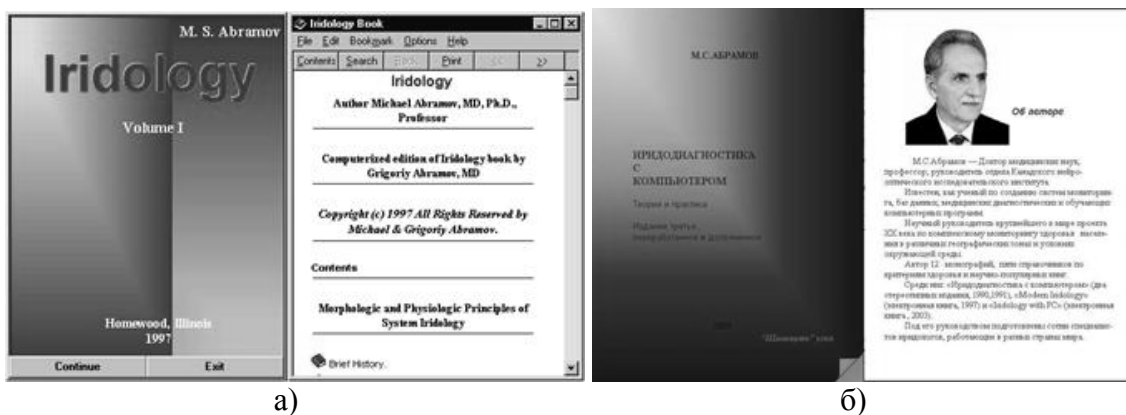


Рис. 29. Электронные учебники «Moder Iridology» (а) и «Иридодиагностика с компьютером» (б) М.Абрамова (США)

Электронный атлас Брайена Марчия (Канада). Электронный атлас на CD «Конституциональные типы» (рис. 30) разработан в канадском нейрооптическом институте Брайена Марчия (Bryan K. Marcia).



Рис. 30. Электронный справочник «Конституциональные типы» (Канада)

Атлас, реализованный в форме pdf-файла, содержит фотографии и подробный текстовый комментарий по 33 конституциональным типам и подтипам радужных оболочек. Разделы атласа:

- лимфатическая конституция и диатез;
- гематогенная конституция;
- билиарная конституция или смешанный тип и диатез;
- патологические конституции;
- конституциональные синдромы;
- конституциональная онконастороженность;
- приложения.

Электронный учебник Тони Миллер (Австралия). Электронный учебник «Iridologist» Тони Миллер (Toni Miller), выпущен австралийской фирмой Naturopathic Technologies (рис. 31).

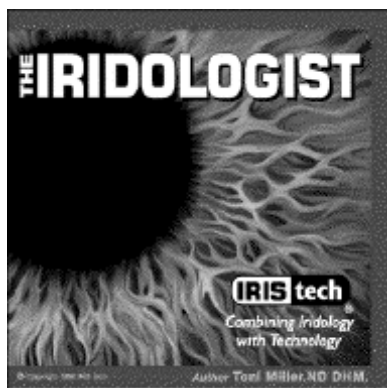


Рис. 31. Электронный учебник «Iridologist» (Австралия)

Тони Миллер – одна из первых специалистов-иридологов Австралии – начала свою деятельность как иридолог в 1983 году. Она основала направление, названное «Интегрированная иридология», в котором попыталась объединить американский (Bernard Jensen) и европейский (Angerer, Deck, Kriege) подходы к изучению радужной оболочки. Клиника Тони Миллер находится в Лэйк Хэвен (Lake Haven) на центральном побережье австралийского штата Новый Южный Уэльс (New South Wales).

2.8.5. Телекоммуникационные средства

С развитием компьютерных сетей развиваются и методы дистанционной диагностики. Одним из первых проектов такого рода в иридодиагностике стал проект канадского нейро-оптического института, руководимого Брайаном Марчия (Bryan K. Marcia, Canadian Neuro-Optic Research Institute) [8]. Технология такой диагностики выглядит следующим образом: пациент с помощью Интернет пересылает в институт фотографии своих радужек, эти фотографии анализируются специалистами института и на оборудовании института, после чего результаты диагностики возвращаются пациенту также с помощью Интернет. Учитывая высокую скорость пересылки информации по Интернет, можно сказать, что в идеале время диагностики будет практически таким же, как если бы пациент находился рядом с врачом. Что касается качества такой диагностики, оно вряд ли будет столь же высоким из-за отсутствия возможности собеседования и прямого наблюдения пациента.

При наличии свободного доступа к Интернет, подобные телекоммуникационные системы могут быть внедрены и в Украине. Схема обмена информацией при этом весьма простая: цифровые фотоснимки радужек пересылаются из школ в районные Центры здоровья, специалисты-иридологи которых оперативно производят дистанционную иридодиагностику с помощью имеющихся в их распоряжении АРМ'ов иридолога.

2.9. Оборудование типового АРМ иридолога

Состав оборудования типового АРМ'а иридолога приведен на рис. 32, а примерный внешний вид оборудования показан на рис. 33.

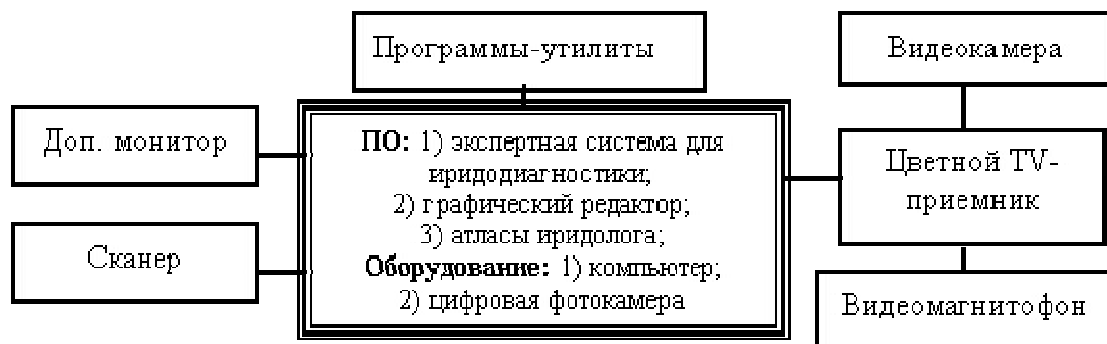


Рис. 32. Состав оборудования типового АРМ'а иридолога



Рис. 33. Внешний вид оборудования АРМ'а иридолога

Ядром оборудования являются компьютер и цифровая фотокамера. В подавляющем большинстве случаев (практическая иридодиагностика в небольших диагностических центрах или школах) таким ядром, из соображений экономии, можно ограничиться. В частности, именно так поступили при создании в 2002 году АРМ иридолога для целей профилактической диагностики школьников Киевского Финансового лицея [12]. При этом использовалась цифровая камера Casio QV-8000 SX (рис. 34).



Рис. 34. Цифровая фотокамера Casio QV-8000 SX

Пример фотографии, полученной с помощью цифровой камеры Casio QV-8000 SX, приведен на рис. 35.

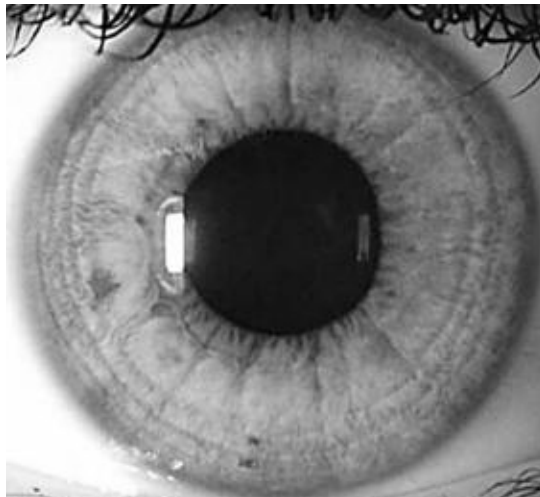


Рис. 35. Фотография радужки, полученная фотокамерой Casio QV-8000 SX

На этом снимке отчетливо наблюдается волокнистая структура радужки, нервные кольца, оттенок пигментации как радужки в целом, так и ее отдельных участков. Если АРМ предназначается еще и для исследовательских целей, в его состав могут быть включены:

- дополнительный монитор;
- сканер;
- видеокамера;
- видеомагнитофон;
- телевизор.

Дополнительный монитор удобно использовать для отображения цветных изображений радужных оболочек глаз, в то время как на основном мониторе отображается графический интерфейс диагностической программы, а также интерфейсы вспомогательных программ, таких, например, как электронные атласы иридолога.

Сканер может понадобиться, если иридолог пользуется обычной пленочной, а не цифровой, фотокамерой. Тогда цветные фотоснимки могут быть введены в память компьютера с помощью сканера (результат такого сканирования приведен на рис. 36).

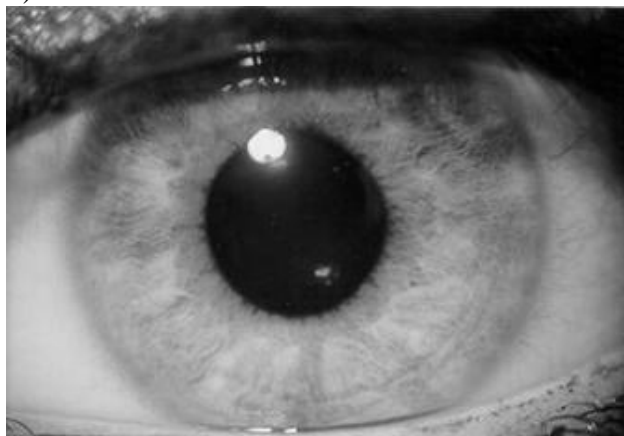


Рис. 36. Фотография радужки, полученная фотокамерой Zenit TTL

Телевизор и видеомэгноффон можно использовать, если иридолог располагает видеокамерой. При этом, вообще говоря, можно и не вводить полученное видеокамерой изображение в компьютер – достаточно наблюдать и анализировать его на экране обычного бытового телевизионного приемника.

Если возникает желание ввести информацию с выхода видеокамеры в компьютер, следует помнить, что современные видеокамеры бывают двух типов: аналоговые и цифровые. Оцифровку сигнала с выхода видеокамеры целесообразно производить с помощью видеокарты компьютера, содержащей функцию ввода видеосигналов (заметим, что далеко не всякая видеокарта обладает такой возможностью). Разумеется, для такого ввода видеосигнала в компьютер необходимо располагать специальными программами-утилитами (обычно поставляемыми в комплекте с драйверами видеокарты). Электрическое соединение видеокамеры и видеокарты осуществляют с помощью специальных кабелей.

Значительно проще перенос информации в компьютер с цифровой видеокамеры. Поскольку оцифрованный видеокамерой сигнал хранится в модуле памяти видеокамеры в виде файла, копирование его в память компьютера производится стандартными средствами ОС Windows. При этом может понадобиться предварительная установка на компьютер специальных программ-драйверов. Электрическое соединение видеокамеры и компьютера при этом осуществляется с помощью специального кабеля через один из последовательных портов компьютера (например, через USB-порт).

Окончательное конфигурирование АРМ'а следует производить, исходя из решаемых задач и финансовых ограничений.

Максимально полный состав оборудования: 1) персональный компьютер; 2) цифровая фотокамера (или цифровая видеокамера); 3) цветной телевизор; 4) видеомэгноффон; 5) цветной принтер; 6) сканер; 7) дополнительный монитор.

Центральными компонентами оборудования являются компьютер и цифровая фотокамера. Если речь идет о создании простейшего АРМ'а для практикующего иридолога или для школьной (профилактической) иридодиагностики, таким ядром из соображений экономии можно и ограничиться.

Если говорить об оборудовании входящего в состав АРМа, то в принципе, сегодня нет необходимости вести собственные разработки оборудования АРМ'а иридолога, поскольку задача ввода в компьютер изображений с выхода видеокамеры легко решается современными мощными видеокартами, а ввод изображений с выхода цифровой фотокамеры осуществить еще легче, особенно при наличии USB-порта.

2.10. Проектирование иридологического ПО

Прежде чем работать с радужной оболочкой нужно ее выделить. Этапы определения радужной оболочки:

1. Найти зрачок (внутренняя часть радужки).
2. Найти границу радужки и белка (внешняя часть радужки).
3. Найти верхнюю и нижнюю веку.

После того как будут проделаны эти действия можно будет приступать к работе с изображением – искать иридопризнаки, определять конституцию человека, ставить диагноз и т.д.

2.10.1. Поиск зрачка

На сегодняшний момент существует достаточное количество методов, которые так или иначе идентифицируют ирис (этот термин широко используется в биометрии и обозначает радужную оболочку). Определение или выделения зрачка наиболее часто используется для наблюдения за вертикальным или горизонтальным положением глаза [15], [16], [17]. Большинство из начальных систем выделения зрачка используют слишком упрощающие предположения, что зрачок представляет собой круг и его центр можно рассчитывать как пересечение соответствующих горизонтали и вертикали. На практике, даже круглый зрачок принимает эллиптическую форму при нецентральной позиции глаза. D.Zhu, S.T.Moore и T.Raphan предложили использовать криволинейные характеристики контура зрачка и уместить их в эллипс [15]. Большинство из предложенных на сегодняшний день методов определения зрачка глаз не предусматривают использование помехоустойчивых алгоритмов.

Некоторые методы, например Wildes, используют специальное оборудование для захвата изображения, чтобы полученное изображение глаза было высокого разрешения, с хорошей контрастностью, освещением (при этом человек, которого снимают, не должна испытывать дискомфорта от слишком яркой вспышки), и отцентрировано (радужка должна находиться в центре изображения). Кроме того, система камер должна быть неинвазивная, то есть не заставлять человека сесть в определенную позу на фиксированном расстоянии от камеры при специальном освещении. Для этого Wildes предлагает специальную систему камер.

Иногда, кроме снимка в видимом диапазоне, делается дополнительный снимок инфракрасной камерой.

Для того, чтобы отделить ирис от остальных деталей на изображении, в простейшем случае можно использовать выделение краев (путем анализа первой производной) и последующую аппроксимацию границ радужки простыми геометрическими объектами. Так, окружность зрачка и внешнюю границу радужки можно найти с помощью преобразования Хафа (Hough transform) [18]. Другие методы дополнительно определяют границу радужки и зрачка двумя параболой, как Wildes, или просто отрезают те части картинки, которые могут не относиться к радужке глаза, как Daugman, Ma [18], [19].

Если для захвата изображения не было использовано специальной аппаратуры, может потребоваться предварительное подавление нежелательных эффектов, таких как отблеск внутри зрачка от вспышки или другого яркого источника света, если эти артефакты мешают корректной работе алгоритма выделения радужки [20]

Для определения зрачка предлагаю использовать метод усиления простых классификаторов, который называется AdaBoost. Усиление простых классификаторов – подход к решению задачи классификации, путём комбинирования примитивных «слабых» классификаторов (рис. 37) в один «сильный» [21]. Под «силой» классификатора в данном случае понимают

эффективность (качество) решения задачи классификации [22]. Слабый классификатор имеет вид:

$$h(x, f, p, \Theta) = \begin{cases} 1, & pf(x) < p\Theta \\ 0, & \text{иначе} \end{cases}$$

где f – признак, p – полярность, которая указывает направление неравности, Θ – пороговое значение.

Финальный сильный классификатор:

$$C(x) = \begin{cases} 1, & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0, & \text{иначе} \end{cases}$$

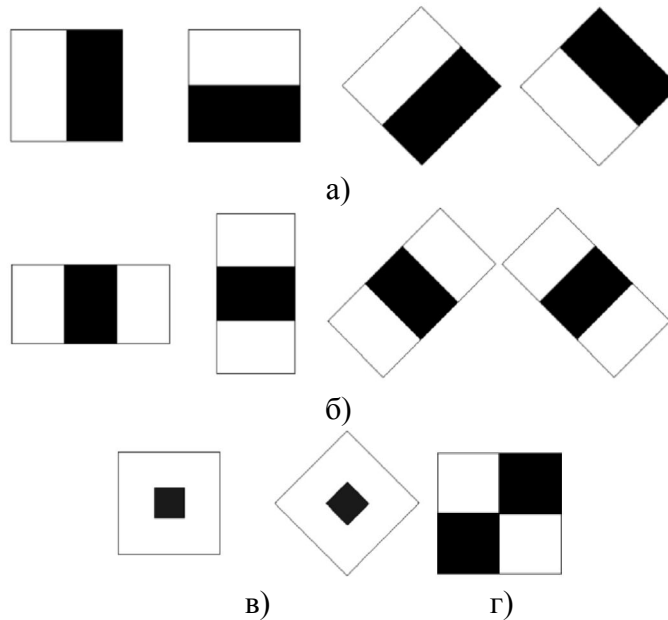


Рис. 37. Виды признаков, которые используются для анализа изображения:
а) граничные; б) линейные; в) центральные; г) диагональные

Таким образом, чтобы распознать радужную оболочку глаза сначала создается набор для классификации. Каждый следующий классификатор строится по объектам, неверно классифицированным предыдущими классификаторами. AdaBoost вызывает слабый классификатор в цикле $t = 1, \dots, T$. После каждого вызова обновляется распределение весов, которые отвечают важности каждого из объектов обучающего множества для классификации. На каждой итерации веса каждого неверно классифицированного объекта возрастают (или аналогично, вес каждого корректно классифицированного объекта уменьшается), таким образом, новый классификатор «фокусирует своё внимание» на этих объектах. Схематически данный алгоритм представлен на рис. 38 в виде управляющей сети, одной из модификаций сетей Петри [23], [24].

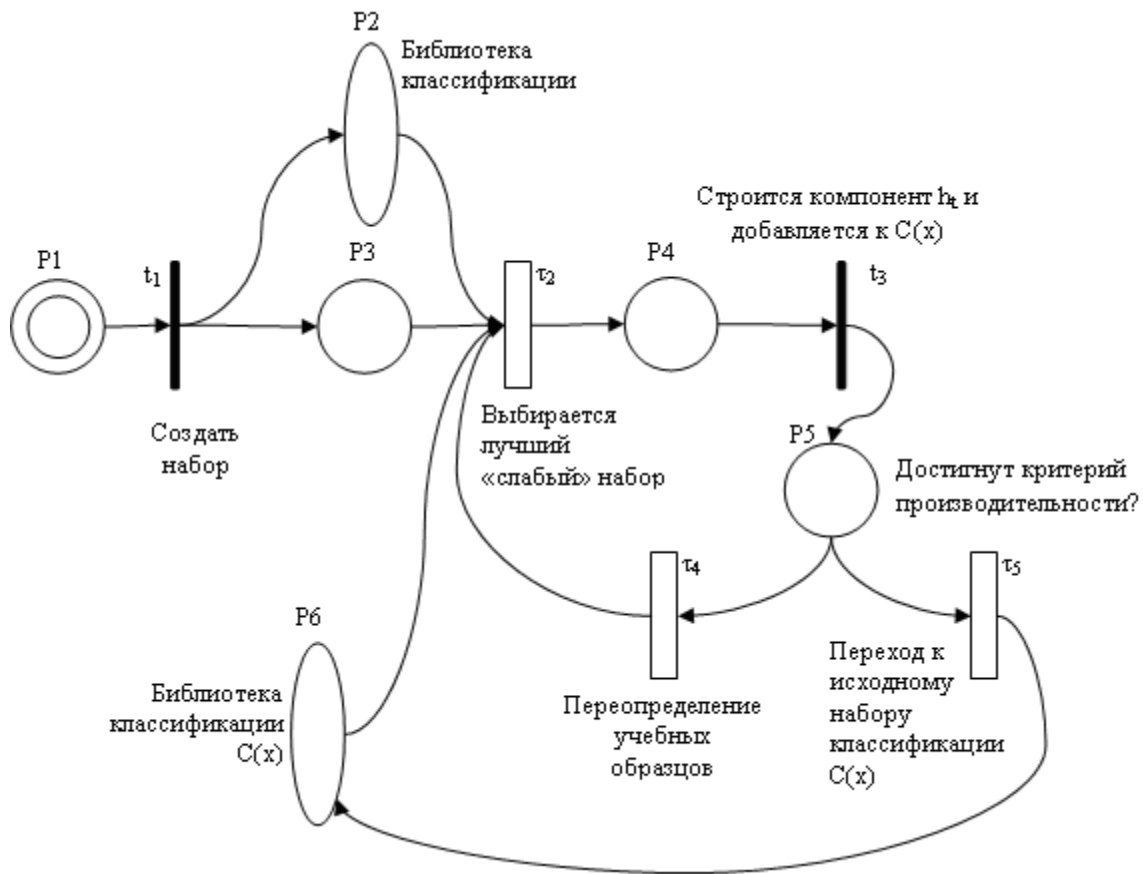


Рис. 38. Алгоритм распознавания методом AdaBoost

Сетью Петри называется ориентированный двудольный (бихроматический) граф $G_p = (P, \Gamma, K, S)$, в котором P и T – два множества вершин, причем $P = \{p_e\}$ – множество вершин мест p_e , $T = \{t_{ij}\}$ – множество вершин переходов t_{ij} ($P \cap T = \emptyset$), соединенных между собой дугами $K = P \rightarrow T \cup T \rightarrow P$ по функциональным правилам S [25].

В общем случае функциональные правила $S = (L_Q, I_R, m, b, w, L)$, где $I_Q: K_Q \rightarrow Q \cup P \rightarrow T$ и $I_R: K_R \rightarrow R \cup T \rightarrow P$ – функции инцидентности; $b: T \cup P \rightarrow L$ – функция обозначения; $w: K \rightarrow \mathbb{N}$ ($\mathbb{N} = 0, 1, 2, \dots, n$) – весовая функция; $m: P \rightarrow \mathbb{N}$ – разметочная функция; $L = X \cup Y$ – используемый алфавит. В частном случае S можно рассматривать как отображение множества дуг, соединяющих вершины P и T , на множество K ($S: (P \rightarrow T \cup T \rightarrow P) \rightarrow K$).

Множества $P = \{p_1, p_2, \dots, p_n\}$ и $T = \{t_1, t_2, \dots, t_n\}$ имеют мощность $|P| = n$ и $|T| = k$. Вместо $P = \{p_1, p_2, \dots, p_n\}$ и $T = \{t_1, t_2, \dots, t_k\}$ часто пишут $P = \{p_e\}$ и $T = \{t_j\}$, $e \in E = \{1, 2, \dots, n\}$, $j \in J = \{1, 2, \dots, k\}$.

В зависимости от функциональных правил S формируется требуемая интерпретация сетей Петри, обуславливающая ее конкретное применение. Обычно в сетях Петри местам p_e соответствуют состояния z_e , а переходам t_j – операции или микрокоманды a_j .

Результаты применения этого алгоритма к изображению глаза показаны на рис. 39.



Рис. 39. Результаты применения алгоритма AdaBoost к изображению глаза

Как видно с рис. 39 метод AdaBoost выдает координаты зрачка стабильно и безошибочно, но, к сожалению, не точно. Это связано с тем, что данный метод преимущественно применяется для распознавания и описания плоских примитивов, анализа движения и отслеживание объектов, обнаружения объектов на изображении и т.д., то есть применяется в тех случаях, когда необходима стабильная, быстрая но не точная (до пикселя) работа. Для уточнения внутренней и поиска внешней границы ириса предлагаю использовать оператор Даугмана.

2.10.2. Интегро-дифференциальный оператор Даугмана

Даугман использует интегро-дифференциальный оператор для определения расположения радужки, а также дуг верхних и нижних век. Интегро-дифференциальный оператор определяется как [10]:

$$\max_{(r, x_0, y_0)} \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r, x_0, y_0} \frac{I(x, y)}{2\pi r} ds \right|$$

где $I(x, y)$ – это изображение глаза, r – искомый радиус, $G_\sigma(r)$ – функция сглаживания по Гауссу, s является контуром круга, который задается как r, x_0, y_0 . Оператор ищет круговую траекторию, где есть максимальное изменение значений пикселей, путем изменения радиуса и координат центра x и y кругового контура. Оператор применяется итеративно. Веки локализируются похожим образом путем контурного интегрирования измененного с кругового на дуговое [11].

Интегро-дифференцирование можно рассматривать как вариацию преобразования Хафа, поскольку оно также использует первые производные изображения и ищет геометрические параметры [11]. Так как оно работает с производной информацией, оно не страдает от проблем с порогом (преобразование Хафа). Однако, алгоритм может неправильно работать с шумом на изображении, например блик, так как он работает только на одном уровне.

2.10.3. Удаление шума

Алгоритм удаления шума состоит из четырех этапов [4]:

- сортировка точек в порядке возрастания их яркости;
- удаление «белого» шума;
- сортировка точек в порядке убывания их яркости;
- удаление «черного» шума.

Обработка точек выполняется двукратным сканированием с использованием дополнительного массива длиной равной количеству различных уровней яркости [12].

Удаление «белого» шума осуществляется при первом сканировании упорядоченных точек. Пусть t_i будет пикселем с яркостью $v(t_i)$. Обработка пикселя состоит из проверки существует ли «белый» шум в области G , содержащей этот пиксель. Сначала область содержит только t_i . Затем область растет так, что некоторые пиксели t' входят в нее, если:

- t' соседствует с областью G ;
- $v(t') \geq v(t_i)$.

Регион растёт, пока не выполнится одно из следующих условий:

- размер полученной области G превышает определенный размер. В этом случае область G не рассматривается как шум и обрабатывается следующий пиксель;
- пиксель не может быть включен в G . В этом случае G является шумом и яркость всех его точек составляет максимальную яркость соседних пикселей [4].

Удаление «черного» шума имеет тот же алгоритм что и удаление «белого» шума.

2.10.4. Уточнение границ

Так как для определения зрачка используется достаточно надежный и «сильный» алгоритм AdaBoost, потому нет нужды применять оператор Даугмана в чистом виде для уточнения границ радужки. Как говорилось ранее, оператор Даугмана пытается вычислить разницу значение уровня серого, между внутренними и внешними пикселями радужки. Даугман показал, что это значение является наибольшим среди всех возможных [10].

У нас есть восемь пикселей окрестности так, как показано на рис. 40. Эти пиксели представляются, как $P_1..P_8$. Функция пытается вычислить значение разницы вычитанием противоположных пикселей. Противоположные пиксели: правые и левые – (P_1, P_5) , верхние и нижние – (P_3, P_7) , и две пары диагональных пикселей – (P_2, P_6) и (P_8, P_4) . Функция игнорирует пиксели на контуре круга пока внешние и внутренние пиксели контура одинаковы. P_1 , P_2 , P_3 и P_8 находятся за пределами, а P_5 , P_6 , P_7 и P_4 – находятся внутри.

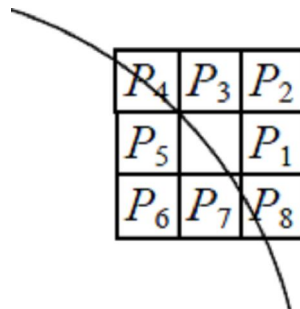


Рис. 40. Внешние и внутренние пиксели радужки

Рис. 41 показывает проблему этого предположения. Функция разности внешних и внутренних точек на рис. 41 окрашена. Если сосредоточиться на Октаде5 – серый прямоугольник в действительности внутри круга, в то время как он должен быть за его пределами.

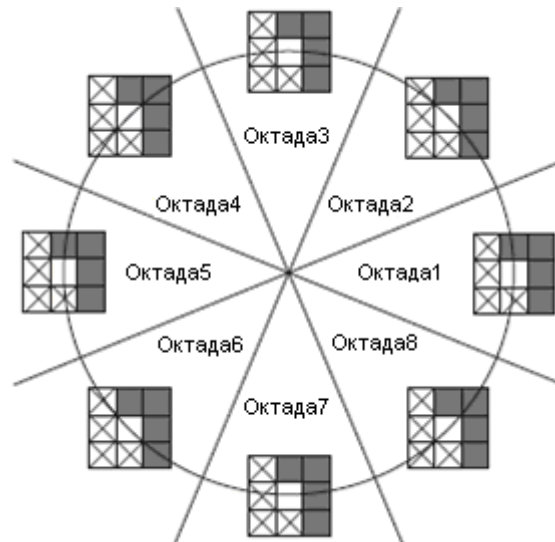


Рис. 41. Неправильные настройки

Правильная настройка внешних и внутренних точек вокруг контура круга изображена на рис. 42.

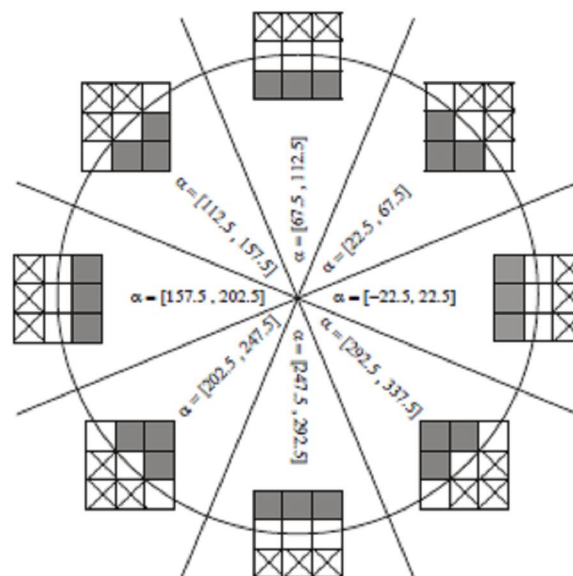


Рис. 42. Правильные настройки пикселей

У нас есть только восемь точек вокруг главного пикселя. Таким образом, мы разделяем круг на восемь регионов, так как внешние и внутренние состояния точек в каждом регионе одинаковы. Это показано на рис. 42. Каждый регион занимает $\frac{2\pi}{8}$ или $\frac{\pi}{4}$, что равно 45° . Центр каждого региона представлен в виде θ_j с $22,5^\circ$ смещением следующим образом:

$$\theta_j = \left(\left[\frac{\alpha_j + 22.5}{45} \right] \times 45 \right)$$

Предлагаются новые коэффициенты матрицы с измерением 3×3 . Матрица умножается на соседние основные точки. Значения коэффициентов матрицы

выбирается из значений $(-1,0,1)$. Коэффициенты пересчета для внешних пикселей равны 1, а для контура и внутренних точек – 0 и -1 соответственно. Путем добавления значения пикселей, после применения коэффициентов, мы получаем реальную разницу между внешними и внутренними пикселями. Коэффициенты матрицы:

$$M = \begin{bmatrix} \sin\left(\left(\theta_j - 45\right) \times \left(\frac{2\pi}{360}\right)\right) & \sin\left(\left(\theta_j\right) \times \left(\frac{2\pi}{360}\right)\right) & \sin\left(\left(\theta_j + 45\right) \times \left(\frac{2\pi}{360}\right)\right) \\ \sin\left(\left(\theta_j - 90\right) \times \left(\frac{2\pi}{360}\right)\right) & 0 & \sin\left(\left(\theta_j + 90\right) \times \left(\frac{2\pi}{360}\right)\right) \\ \sin\left(\left(\theta_j - 135\right) \times \left(\frac{2\pi}{360}\right)\right) & \sin\left(\left(-\theta_j\right) \times \left(\frac{2\pi}{360}\right)\right) & \sin\left(\left(\theta_j + 135\right) \times \left(\frac{2\pi}{360}\right)\right) \end{bmatrix}$$

2.10.5. Определение иридопризнаков

После того, как станет известно месторасположение «бублика» ириса – можно будет проводить дальнейшую его обработку, которая включает в себя поиск иридопризнаков. Для поиска иридопризнаков предлагаю вначале отфильтровать изображение с помощью правил построенных на основе цветовых пространств. Цветовое пространство – это математическое представление набора цветов. Существуют три наиболее популярные цветовые модели – это RGB (используется в компьютерной графике); YIQ, YUV или YCbCr (используются в видео системах); и CMYK (используется в полиграфии).

RGB – аддитивная цветовая модель, как правило, описывает способ синтеза цвета для воспроизведения цвета. Выбор основных цветов обусловлен особенностями физиологии восприятия цвета сетчаткой человеческого глаза.

YUV – цветовое пространство, которое используется как составляющая цветного видео стандарта PAL, NTSC и SECAM. Черно-белые системы используют лишь яркостную информацию (Y); цветная информация содержится в компонентах U и V.

$$Y = 0.299R + 0.587G + 0.114B$$

$$U = -0.147R - 0.289G + 0.436B = 0.498(B - Y)$$

$$V = 0.615R - 0.515G - 0.100B = 0.877(R - Y)$$

YIQ – цветовое пространство полученное из цветового пространства YUV и опционально использовалось NTSC стандартом. I означает «синфазный» (inphase), а Q – «квадратура» (quadrature), указывая на метод модуляции, используемый для перевода цветовой информации.

$$Y = 0.299R + 0.587G + 0.114B$$

$$I = 0.596R - 0.275G - 0.321B = V \cos 33^\circ - U \sin 33^\circ = 0.736(R - Y) - 0.268(B - Y)$$

$$Q = 0.212R - 0.523G + 0.311B = V \sin 33^\circ + U \cos 33^\circ = 0.478(R - Y) + 0.413(B - Y)$$

Цветовое пространство YCbCr было разработано как часть ITU-R BT.601 при разработке мирового цифрового компонентного видео стандарта. YCbCr определяется как смещенная версия YUV. Существует несколько версий данного цветового пространства: SDTV и HDTV

$$Y_{601} = 0.299R + 0.587G + 0.114B$$

$$Cb = -0.172R - 0.339G + 0.511B + 128$$

$$Cr = 0.511R - 0.428G - 0.083B + 128$$

$$Y_{709} = 0.213R + 0.715G + 0.072B$$

$$Cb = -0.117R - 0.394G + 0.511B + 128$$

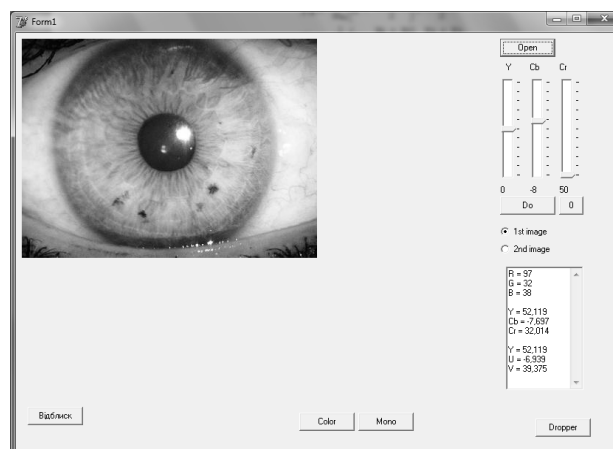
$$Cr = 0.511R - 0.464G - 0.047B + 128$$

Используя данные цветовые пространства, можно построить любые классификаторы под любые потребности классификации объектов наблюдения на изображении. Так, в ходе исследований было определено следующее правило для фильтрации изображения, которое строится на основе цветового пространства YCbCr: $(Y = 0) \text{ and } (Cr < -8) \text{ and } (Cb > 50)$ [24]. В итоге применения этого правила, можно получить битовое изображение с выделенными иридопризнаками. Результаты применения фильтра к изображениям продемонстрированы на рис. 43.

В дальнейшем данное изображение стоит сегментировать и спроецировать на иридосхему.

2.10.6. Анализ плотности ириса

В иридологии важным показателем являются не только иридопризнаки, но и плотность ириса (см. п. 2). Для анализа плотности ириса предлагаю использовать метод «Локальных бинарных шаблонов» (ЛБШ). ЛБШ определяется для каждого пикселя в качестве порога для соседних пикселей, а также в виде бинарной строчки бит. ЛБШ представляет собой описание очертания пикселя изображения в двоичной форме. Оператор ЛБШ, который применяется к пикселю изображения, использует восемь точек очертания, принимая центральный пиксель в качестве порога. Пиксели, которые имеют значение больше, чем центральный пиксель (или равны ему), принимают значение «1», те, которые, меньше центрального, принимают значение «0». Таким образом, получается восьмиразрядный двоичный код, описывающий очертание пикселя. Затем данные коды переводятся в десятичную систему и формируют гистограмму изображения. Каждый ЛБШ-код представляет тип микроизображения структуры, а их распределение можно использовать в качестве описания текстуры (рис. 44) [22], [24].



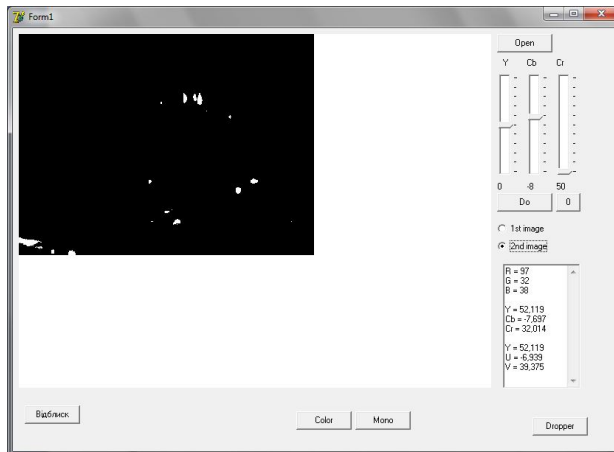
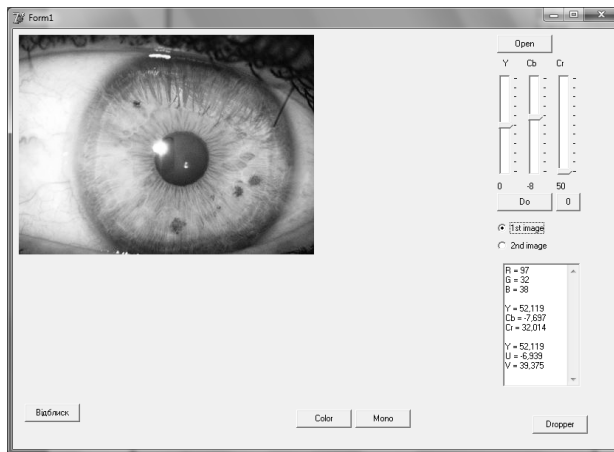
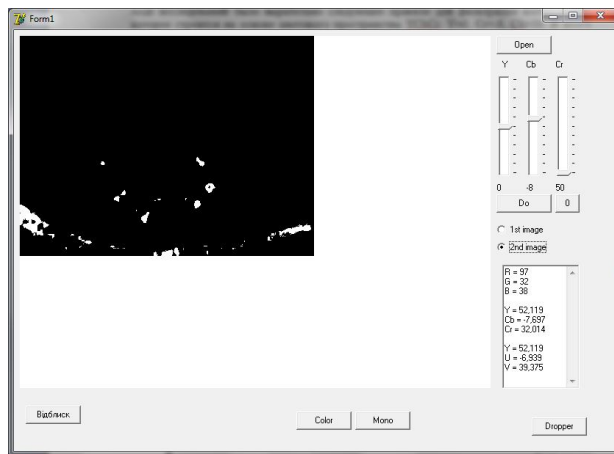


Рис. 43. Применение фильтра на основе YCbCr

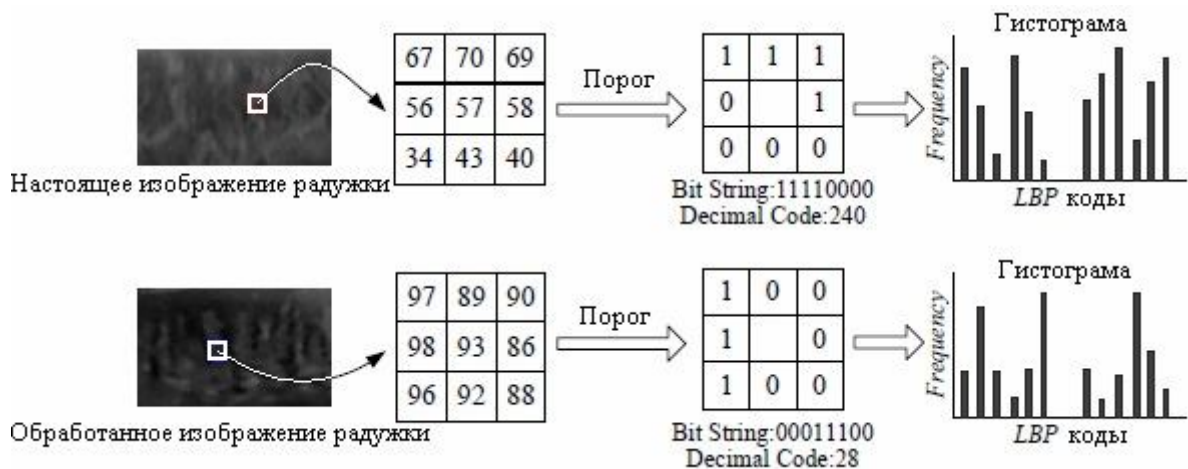


Рис. 44. Использование метода ЛБШ для описания текстуры

Заключение

В данной работе проводится обзор иридодиагностики, как науки, и факторов, которые формируют выводы в иридодиагностике: цвет глаз, плотность радужки и ее география. Также было проанализировано необходимое и дополнительное оборудование для проведения иридодиагностики. Необходимое оборудование предназначено только для проведения иридодиагностики и постановки диагноза. Дополнительное оборудование, которое включает дополнительный монитор, сканер и т.д., необходимо для более широкого изучения вопросов иридодиагностики и предназначено, в основном, для исследований в этой области. Был проведен обзор существующих программ и систем для иридодиагностики. На основании этого был сделан вывод о целесообразности создания кардинально нового программного обеспечения, которое бы обладало свойствами графического редактора, экспертной системы и в автоматическом режиме осуществляло поиск радужной оболочки и иридоознак в едином комплексе. Были рассмотрены преимущества и недостатки существующих алгоритмов и схем локализации радужки, нормализации изображения, подавление помех, а также использование цветных пространств для локализации иридопризнаков на радужной оболочке глаза.

Список литературы

1. Вельховер Е.С. Клиническая иридология. – М.: «Орбита», 1992. – 431 с.
2. Кривенко В.В., Лисовенко Г.С., Потєбня Г.П., Сядро Т.А. Иридодиагностика. Справочник. – К.: «Украинская Советская Энциклопедия» имени М.П.Бажана, 1991. – 140 с.
3. Продеус А.Н., Сядро Т.А. Особенности разработки программы ESID-3 для компьютерной иридодиагностики // Электроника и связь – №9 – 2000. – С.74-77.
4. Продеус А.Н. К разработке нового поколения систем компьютерной поддержки иридолога // Электроника и связь – № 6, том 1 – 1999 – с. 194-198.
5. Продеус А.Н., Захрабова Е.Н. Экспертные системы в медицине. Учебное пособие. – К.: ТОО «ВЕК+», 1998. – 326с.
6. Петенко О.В., Гречишникова Н.И. Практическое пособие по иридодиагностике // Душанбе: «ИРФОН», 1991. – 451с.
7. Продеус А.Н., Сядро Т.А. Программа ESID-3 для компьютерной иридодиагностики // Доклады IV международной научно-практической конференции по народной и нетрадиционной медицине. – К., 2002.
8. Продеус А.Н., Сядро Т.А. Компьютерная иридодиагностика – современное состояние // Электронный атлас иридолога. Часть 1. Иридогенетические синдромы. Учебное пособие на CD-ROM. – К., 1999-2002.
9. Продеус А.Н., Сядро Т.А. Проблемы реализации байесовской стратегии при автоматизации иридодиагностики // Электроника и связь – №2, часть II – 1997. – С. 34-36.
10. Продеус А.Н. Компьютерная программа «ESID-2» для иридодиагностических исследований // Народна та нетрадиційна медицина України, №1(2), Київ, 1997. – С. 9-12.
11. Продеус А.М., Сядро Т.А., Верба А.І., Зафіріді О.Г., Мартиненко В.О. Свідоцтво про державну реєстрацію прав автора на твір, ПА#262. Комп'ютерна програма «ESID-2». Дата реєстрації 14 жовтня 1996 р.
12. А.Н.Продеус, Т.А.Сядро, Ю.В.Шукевич. АРМ школьного врача-иридолога // Труды международной дистанционной научно-практической конференции «Автоматизированное рабочее место врача». Днепропетровск, 2002г.
13. Продеус А.Н., Сядро Т.А., Шукевич Ю.В. Компьютерная иридодиагностика в контексте валеологизации школьного образования // Труды международного конгресса «Единый мир – здоровый человек», Крым-Ялта, 2004 г.
14. Абрамов М.С. Иридодиагностика с компьютером // Ташкент, изд. им. Ибн Сины, 1991. – 194 с.
15. Zhu D., Moore S.T., Raphan T. Robust pupil center detection using a curvature algorithm // Computer methods and programs in biomedicine. – 1999. –№3 (59). – С. 145–157.
16. Dongheng L., Winfield D. and Parkhurst D.J. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches // Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2005). – San Diego, USA, June 2005. – Vol. 3. – pp. 79.

17. Moore S.T., Haslwanter T., Curthoys I.S. and Smith S.T. A geometric basis for measurement of threedimensional eye position using image processing // *Vision research*. – 1996. – Vol. 36. – pp. 445-459.
18. Li Ma, Yunhong Wang, Tieniu Tan. Iris Recognition Using Circular Symmetric Filters // *Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02)* – 2002. – pp. 20414-20418
19. Richard P. Wildes. Iris Recognition: An Emerging Biometric Technology, *Proceedings of The IEEE*, vol. 85, no. 9 – September 1997 – pp. 1344-1347.
20. Christel-loic Tisse, Lionel Martin, Lionel Torres, Michel Robert. Person identification technique using human iris recognition // *Proceedings of Vision Interface* – 2002. – pp. 294-299.
21. Кузьмук В.В., Волошин М.В. Визначення впливу моделі опису об'єкта на достовірність його ідентифікації в системах комп'ютерного зору // *Засідання третьої міжнародної науково-технічної конференції МЕЕС'10* – 2010. – С. 69-72.
22. Волошин М.В. Моделі опису об'єкта та достовірність ідентифікації в системах комп'ютерного зору // *Восточно-Европейский журнал передовых технологий*. – 2010. – №4/7 (46) – С. 56-63.
23. Кузьмук В.В., Супруненко О.О. Модифицированные сети Петри и устройства моделирования параллельных процессов: Монография. – К.: Маклаут, 2010. – 252 с.
24. Волошин Н.В., Кузьмук В.В., Тараненко Е.А. Моделирование и распознавание информативных участков в автоматизированных системах ириодиагностики // *Восточно-Европейский журнал передовых технологий*. – 2011. – №2/2 (50) – С. 65-69.
25. Васильев В.В., Кузьмук В.В. Сети Петри, параллельные алгоритмы и модели мультипроцессорных систем. – К.: Наукова думка. 1990. – с. 209.
26. John Daugman, High Confidence Visual Recognition of Persons by a Test of Statistical Independence // *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, p. 1148–1161, November 1993.
27. Волошин Н.В. Метод определения зрачка в автоматических системах ириодиагностики // *Вестник Национального технического университета «ХПИ»*. – 2011. – №10 – С. 52-55.

РОЗДІЛ 3. Моделі та засоби комп'ютерної діагностики енергетичних процесів в організмі людини (Єремєєв Б.М.)

3.1 Біорезонансна діагностика і терапія

Багатство ритмів електромагнітних коливань, виявлених у зовнішньому середовищі, відповідає і розмаїттю в організмі людини. Однією з основних характеристик біологічних систем є присутність хвильових спектрів, які є складовими біофізичних, фізіологічних, біохімічних процесів, а іноді й визначають їх спроможність.

Тривалі спостереження за природними явищами та особливостями реакцій людського організму дозволили філософам і лікарям сходу створити систему енергетичних каналів і взаємозв'язків між органами та системами, які успішно застосовуються в рефлексотерапії.

Процес діагностики стану пацієнта на основі оцінки енергетичного потенціалу біологічно активних центрів, що дозволяє виявити патологічні процеси та їх збудники. Вважається, що при порушенні розвитку патологічних процесів однією з перших порушується гармонія хвильових процесів. Це проявляється у вигляді пошкодження ритмів управління основними фізіологічними процесами, дисбалансі корково-підкоркових взаємодій на рівні нейрогіпофіза і гіпоталамуса. Порушення керуючої функції нервової системи тягне за собою пошкодження, така, складно регульована система як ендокринна, в імунній системі відмічаються порушення гуморального та клітинного імунітету – ця стадія розвитку патологічних реакцій відповідає рівню функціональних порушень, хронічний перебіг яких веде до виникнення більш важкої тканинної та органної патології.

Сучасні апаратно-програмні забезпечення дозволяють проводити, реєструвати хвильові процеси у вигляді їх частотних характеристик. Після тривалого періоду досконального аналізу та дослідження ефектів взаємодії хвильових процесів у біологічних системах, нарешті, стало можливим й застосування цього наукового досягнення в практичній медицині. Цей напрямок ліг в основу абсолютно нової галузі медицини – біорезонансної діагностики і терапії.

Новий підхід дозволить інтегрувати в єдине ціле модель організму людини, роз'єднану на окремі органи і системи, вузькими спеціалізаціями академічної медицини.

3.1.1 Принцип та методика діагностики людини на основі БАТ і меридіанів

Після того як французьким фахівцем з акупунктури Нібоє було відкрито більш високу електропровідність в області біологічно активних точок (БАТ) в порівняно з оточуючими ділянками шкіри, виникла можливість створити ряд методів та пристроїв для діагностики та терапії на основі енергетичних центрів. Значення електричного потенціалу в БАТ і меридіанах дозволяють дізнатися про стан пацієнта, а вплив на них електричного струму дозволяє досягти терапевтичного ефекту.[3]

Всі діагностичні методи, що використовують електрошкірні вимірювання (ЕКІ) в області меридіанів, для інтерпретації одержуваних значень, використовують взаємозв'язок шкіри і внутрішніх органів на основі спільності сегментарної іннервації. В даному розділі будуть представлені основні методи діагностики стану людини, основані на енергетичному потенціалі біологічно активних точок та меридіан, що використовуються в сучасних системах біорезонансної діагностики.

3.1.1.1 Метод сегментарної діагностики

Сегментарна діагностика використовує факт, який лежить в основі життєдіяльності всіх тварин і людини – принцип сегментарного будови. Під сегментом розуміють обмежені ділянки шкіри (зони), у яких при захворюваннях внутрішніх органів часто з'являється відображення болю, змінюються параметри електричного опору шкіри, а також, зміни чутливості у вигляді больової і температурної гіперестезії. При ураженні внутрішнього органу патологічні імпульси по чутливих нервових волокнах передаються у відповідний сегмент спинного мозку і викликають збудження сегментарного апарату, що включає другі нейрони шкірної чутливості і рухові нейрони (іннервують м'язи). Процеси збудження і гальмування у відповідних сегментах спинного мозку призводять до зміни електрошкірного опору у відповідну дерматому (обмежена ділянка шкіри, яка пов'язана з відповідним сегментом спинного мозку і внутрішнім органом), що відбивається на показниках сегментарного тесту.

В основу даного методу покладено принцип шести-квадрантних відведень по Фоллю (див. таблицю 1) з урахуванням відведень Пфлаума. Вимірювання в даних відведеннях проводять за допомогою ручних (циліндричних), ножних і лобових пластинчастих електродів. Вимірювання в «великих» відведеннях дають попередню інформацію про локалізацію існуючих порушень та їх характер. Так, значення в межах від 82 до 86 одиниць інтерпретують, згідно Р. Фоля, як нормальні; значення понад 86 можуть бути пов'язані з запальними процесами, в той час як при показниках нижче 82 одиниць можна припускати наявність дегенеративних процесів (детальні показники норм для різних органів представлені в додатку А). За допомогою унікальних технічних рішень сучасні системи дозволяють проводити діагностику близько 30 сегментів тіла людини. Кожен сегмент несе інформацію про конкретний орган, систему та ділянку тіла, а також стан сегментарного апарату спинного мозку і вегетативної системи [9].

Дана методика за рекордно короткий час (лише 30 сек.) дозволяє отримати інформацію про всі органи і системи, їх фізіологічний стані та патологічні зміни у них, завдяки величезній селекторній базі програми надає можливість провести тестування на наявність широкого спектру збудників і патологічних станів, визначити індивідуальну схему терапії. Селекторна база АПК, яка складається з більш ніж 35000 тестових резонансних частот, дозволяє виявити паразитарні та інфекційні захворювання, етіологічний фактор і його роль у патогенезі кожного конкретного захворювання, дозволяє здійснити підбір оптимальної схеми терапії, створення індивідуально підбраного за результатами тестування комплексного інформпрепарату та проведення моніторингу процесу лікування [11].

Таблиця 3.1

Техніка проведення та алгоритм інтерпретації показників вимірювань по 6-ти відведенням

№ п/п	Відведення	Інформаційно-функціональні зв'язки
1	чоло зліва – ліва рука	ЦНС; ліве око; ліве вухо; ліва лобна і гайморові пазухи; шийний відділ хребта; мигдалини; верхня щелепа; ліві відділи верхня частка лівої легені; нерви, судини і суглоби лівої руки
2	чоло зліва – чоло праворуч	ЦНС; шийний відділ хребта; лобні пазухи; зуби верхньої щелепи; мигдалини
3	права рука – чоло праворуч	Зуби нижньої щелепи; мигдалини; праве вухо; праве око; шийний відділ хребта; праві відділи серця; верхня частка правої легені; нерви; судини та суглоби правої руки; ЦНС
4	права рука – ліва рука	Серце; легені; стравохід; щитовидна заліза; шийно-грудний перехід хребта; шлунок; діафрагма
5	ліва нога – ліва рука	Ліва нирка; серце; ліва легеня; грудний відділ хребта; стравохід; шлунок; панкреас; сигма; спадний відділ ободової кишки
6	ліва нога –права нога	Урогенітальні органи; пряма кишка; попереково-крижовий відділ хребта
7	права рука – ліва нога	Гепато-біліарна система; апендикс; права нирка; висхідний відділ ободової кишки; поперековий відділ хребта; шлунок; панкреас; тонка кишка

Сегментарна діагностика, як універсальний метод, базується на наступних принципах:

- 1) жоден орган у людському організмі не переверщує шкіру по ширині смуги різноманітних фізіологічних процесів, тому ніякий орган не може краще шкіри представляти динамічні зміни внутрішнього середовища організму;
- 2) завдяки генетичним і еволюційним інформаційно-функціональним зв'язкам між певними біологічними активними зонами шкіри, з одного боку, і внутрішніми системами та органами, з іншого боку,

можливо оцінювати, що відбувається в них: біохімічні, біоелектричні та інші процеси;

- 3) численні дослідження підтверджують наявність залежностей між зміною електричних параметрів у певних зонах шкіри і зміною стану внутрішніх органів і систем [9].

Метод дозволяє:

- 1) проводити збір інформації про функціональний стан організму і його систем;
- 2) визначати функціональний статус організму в цілому і його частин (основних функціональних систем), який представляє собою як функціональні зміни, так і відхилення від норми в процесі саморегуляції;
- 3) проводити контроль ефективності будь-якої терапії;
- 4) визначати оцінку динамічних терапевтичних дій проведенням вимірювань під час і між сеансами терапії;
- 5) створювати бази даних та статистичного аналізу;
- 6) забезпечувати швидку і зручну оцінку дії препарату або будь-якого іншого об'єкта або предмета (прикраси, предмети одягу, продукти харчування і т.д.).

Протипоказання:

- 1) наявність у пацієнта електрокардіостимулятора;
- 2) значні порушення цілісності шкірного покриву [9].

Отже, сегментарна діагностика є перспективним напрямком експрес-діагностики великої кількості пацієнтів, незважаючи на ряд протипоказань, оскільки відсоток їх наявності у хворих становить близько 1%.

3.1.1.2 Метод електроакупунктурної діагностики по Фолю

Метод електроакупунктурної діагностики по Фолю був введений у медицину в 1953 році німецьким доктором медицини Рейнхольдом Фолем та займає провідне місце серед сучасних методів комп'ютерної діагностики. Електроакупунктура по Р. Фолю (ЕАФ) – метод діагностики, який об'єднує в собі основи китайського вчення про акупунктури і можливості сучасної електроніки та обчислювальної техніки [5].

ЕАФ полягає у діагностиці стану пацієнта на основі вимірів опору в біологічно активних точках, використовуючи розроблену ним модель (вимоги до якої представлені в розділі 3.2). Доктору Р. Фолю вдалося не тільки істотно розширити уявлення про систему акупунктурних меридіанів, а й систематизувати точки акупунктури за їх анатомо-інформаційним взаємозв'язком з внутрішніми органами та системами.

Для проведення діагностики використовується вплив на БАТ напругою в діапазоні від 1,2 до 1,5 V та струмом до 20 мкА. В результаті виникає опір внутрішньої електрорушійної сили (ЕРС), його значення та часові характеристики (ефект «падіння стрілки») дозволяють оцінити фізичний стан акупунктурної точки. Послідовно вимірявши таким чином значення БАТ певного меридіану, можна охарактеризувати його стан, виявивши патологічні зміни в енергетиці меридіана [3].

Для виявлення патології при електропунктурному обстеженні враховують наступні параметри: величина показника, величина "падіння" індикатора (стрілки), асиметрія значень показників, швидкість досягнення максимального значення показника

Діапазон вимірювань варіює від 0 до 100 умовних одиниць (Уо) та має наступну «паталого-анатомічну оцінку»: 100-90 – загальне запалення; 90-82 – часткове запалення; 80-66 – загальна збудливість, що веде до стану передумови хвороби; 65-51 – збудження в межах фізіологічної норми; 50 – нормальний стан; 49-40 – початкова стадія дегенерації; 39-30 помірної стадія дегенерації; 29-20 – значна дегенерація; <20 – кінцева стадія дегенерація, атрофія, карцинома; <10 – стан перед смертю [5].

Феномен «падіння» індикатора (стрілки) полягає в зниженні початкового показника електропровідності на більше 5 одиниць умовних одиниць. Зазвичай падіння стрілки відбувається протягом 1-3 секунд, однак при низьких значеннях на БАТ для виявлення падіння стрілки може бути необхідно 30-60 секунд. Р. Фоль та його послідовники розцінюють падіння стрілки як свідчення наявності органічних уражень в органі або системі. Проте, даний феномен може виникати після проведення біорезонансної терапії в режимі «розкачки», а також на ранніх стадіях вживання гомеопатичних лікарських засобів, що також супроводжується підвищенням раніше зниженого показника і поліпшенням клінічного стану. Таким чином, інтерпретація падіння стрілки повинна проводитися з урахуванням аналізу клінічної ситуації, оскільки в деяких представлених вище ситуаціях поява падіння стрілки була етапом у лікувальному процесі, що свідчить про позитивну динаміку. Цей показник служить основним показником порушення функції відповідних органів або систем.

Різниця в показниках у симетричних точках вимірювання меридіанів праворуч і ліворуч, що перевищує 5 одиниць шкали свідчить про можливість одностороннього патологічного процесу. Найбільш часто асиметрія показників виявляється при наявності симптомів вегето-судинної дистонії, травми, при порушенні мозкового кровообігу. Проте, у жінок в дітородному віці в період овуляції на меридіані сечового міхура і меридіані ендокринної системи реєстрували минущу асиметрію показників.

Швидкість досягнення максимальних значень може варіювати від практично миттєвої до декількох секунд. Прискорений підйом корелює з інтоксикацією організму, а повільне збільшення значення інтерпретують як наявність хронічного патологічного процесу.

Варто відмітити, що представлені вище параметри діагностики мають різний рівень значимості при формуванні діагнозу, серед них перші два є параметрами діагностики, що мають найбільшу вагу при формуванні діагнозу.

Важливим досягненням в біорезонансній діагностики стало відкриття доктором Фолем зміни показників електропровідності при внесенні в електричне коло препарату або його інформаційно-енергетичний характеристик, що відповідає змінам в організмі людини після прийняття даних ліків, харчів, косметики та т.і. За рахунок цього можна підібрати індивідуально найбільш підходящий препарат для лікування пацієнта.

Р. Фолем і його послідовниками описано понад 400 точок вимірювання (класичних акупунктурних та відкритих Р. Фолем) для електропунктурної діагностики. Точки вимірювання, як і в класичній акупунктурі, згруповані по

меридіанах, частина з яких використовується для визначення функцій органів (меридіани серця, нирок), частина для визначення функцій систем (меридіани кровообігу, ендокринний), а частина для визначення стану тканини або обміну (меридіан сполучнотканинної і жирової дегенерації). Для електропунктурної експрес-діагностики рекомендується використовувати контрольні точки вимірювання (КТВ). Ці БАТ не входять до числа класичних точок акупунктури, що описані Р. Фолем, і є репрезентативними для даного меридіана, тобто змінюються при порушенні функції всіх органів або систем, що мають відношення до меридіану. Для розширеної електропунктурної діагностики необхідно провести вимірювання по біологічно активним точкам, які були обрані на основі експрес-діагностики, при цьому для більш детальної оцінки обирається меридіан з найбільшим відхиленням від норми. Точки вимірювання по меридіанах, їх функція і локалізація використовуються відповідно до міжнародної акупунктурної номенклатури.

Незважаючи на високу ефективність діагностичного ефекту від 60 до 85% (в залежності від досвіду спеціаліста) та достатньо просту методику, електропунктурна діагностика вимагає значних затрат часу від 1,5 до 2,5 годин. Саме тому його послідовники розробили ряд більш раціональних методик та модифікацій дослідження динаміки параметрів електропровідності БАТ, як для діагностичних так і для лікувальних цілей. На сьогоднішній день до таких методів належать VEGA-тест, вегетативний біорезонансний тест, біорезонансна терапія (БРТ), Мора-терапія, що являються варіантами електроакупунктури по Фолю [5].

Найбільш відомою в Німеччині, а в останні роки і в Росії є «біометрична системна діагностика» (БСД). Автор модифікації – один з провідних німецьких фахівців Хайнріх Россманн. Кількість досліджуваних точок скорочено ним до 122, послідовність вимірів проводиться за заданою схемою, результати вносяться в пам'ять комп'ютера та обробляються. Для медикаментозного тестування використовуються тест-ампули, а для лікування – підібрані тестуванням нозоди у вигляді ін'єкцій та стандартні гомеопатичні препарати. У середньому тривалість сеансу сягає півтори години.

Модифікації електропунктури по Фолю мають різну спрямованість. Варіант, запропонований М. Л. Лупічевим, виключає квадратні виміри, крім тих які проводяться на контрольних точках (2-3 БАТ) на кожному меридіані. Вважається, що такий порядок підвищує діагностичну цінність отримуваних показників. Під час медикаментозного тестування шукається одне гомеопатичний засіб, який нормалізує показники БАТ на всіх меридіанах. Для кожної БАТ формується список препаратів, для нормалізації показників серед яких шукається спільний. На основі представленого методу формування діагнозу здійснюється з урахуванням електропровідності біологічно активних точок, феномену «падіння стрілки» та інших параметрів запропонованих Р. Фолем.

У модифікації доктора Сарчук В. М., квадрантні виміри не проводяться, а дослідження починається з контрольних точок (КТ) усіх меридіанів. За отриманими показниками визначається імовірно найбільш пошкоджений орган або система. Першим пропонується досліджувати меридіан нервової дегенерації. Перш за все, проводиться тестування вірусних нозодів. Для тестування нозодів і гомеопатичних препаратів пропонуються використовувати касети, які містять речовини – носій інформаційно-енергетичних характеристик препарату. Діагноз формується як на підставі показників електропровідності, так і за результатами

тестування нозодів, приготованих з біопсійного патоморфологічного матеріалу. Для лікування використовується біологічно активна рідина (БАЖ) – дистильована вода, оброблена електромагнітним полем, хвильові характеристики якого відповідають показникам комплексу нозодів і гомеопатичних препаратів, нормалізованих при тестуванні показників електропровідності всіх БАТ. Тривалість проведення діагностування по даній методиці варіює в межах однієї години.

Методика розроблена фахівцями фірми Медіса (м. Сімферополь), відрізняється від попередньої тим що дослідження починається з меридіана з'єднувальної тканини – паренхіматозної і епітеліальної дегенерації (ЗПЕД). Результати вказують на область і органи, імовірність поразення яких відносно висока. Виходячи з результатів дослідження СПЕД меридіана, вибирається алгоритм подальшого обстеження. Спочатку вимірюється електропровідність контрольних точках відібраних меридіанів, потім на меридіані з найгіршими показниками в КТ досліджується електропровідність усіх точок і проводиться тестування етіологічних нозодів, патоморфологічних нозодів, гомеопатичних препаратів і т.д.

Вегетативний біорезонансний тест проводиться на 3-5 БАТ. Техніка вимірювань полягає у легкому дотику щупа до шкіри в проекції біологічно активних точок. Отриманий показник прирівнюється до значення 80, що відповідає нормі. Медикаментозне тестування здійснюється використовуючи препаратний «фільтр», при тестуванні біологічно активних точок обирається та, яка змінює показники електричної провідності до норми.

3.1.1.3 Діагностика за методом Накатані

Даний метод діагностики був введений в медицину японським лікарем Накатані в 1956 році. Метод оснований на ідеях східної медицини, зокрема на уявленнях про меридіани. Метод діагностики по Накатані полягає у визначенні загального стану меридіана, шляхом впливу на біологічні точки струмом від 20 до 100 мкА та напругою в діапазоні від 9 до 12 В. На відміну від методу Фоля ця методика використовує більші значення напруги, за допомогою яких ігнорується опір внутрішньої електрорушійної сили. За рахунок використання в якості БАТ репрезентативних (що відображають середнє значення меридіанів в цілому) точок (рис.1) – метод є актуальним для проведення експрес-діагностики загального і енергетичного стану організму. Таким чином, вимірявши невелику кількість точок (таблиця 2), можливо отримати інформацію про стан усіх 12 меридіан [8].

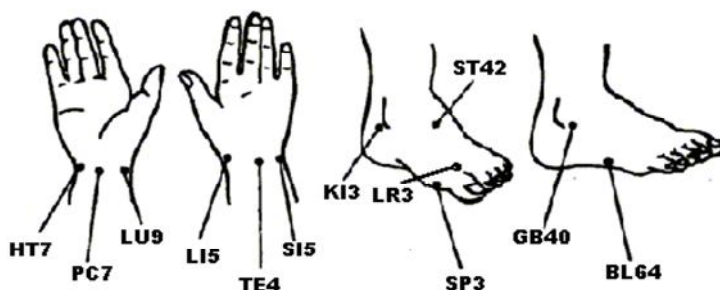


Рис.1 Репрезентативні точки методу Накатані

Накатані вдалося виявити закономірність зміни електричного опору в репрезентативних точках і енергетичного стану меридіана, а відповідно і органу або органів, з ним пов'язаних. Накатані розробив зручну і наочну діагностичну карту, найбільш оптимальну для лікарів, які мислять категоріями китайської акупунктури. Надалі багато дослідників розробляли численні алгоритми розрахунку вимірювань за Накатані, які дозволяли інтерпретувати результати в зручному для європейського лікаря вигляді [4].

Таблиця 2

Репрезентативні точки і порядок електропунктурної діагностики за І.Накатані (рис.1)

Позначення репрезентативних точок	Назва		Приналежність меридіану
Меридіани на руці			
H1	LU9*	P9	легенів
H2	PC7*	MC7	перикарда
H3	HT7*	C7	серця
H4	SI4*	IG4	тонкого кишечника
H5	TE4*	TR4	трьох обігрівачів
H6	LI5*	GI5	товстого кишечника
Меридіани на нозі			
F1	SP3*	RP3	селезінки та підшлункової залози
F2	LR3*	F3	печінки
F3	KI3*	R3	нирок
F4	BL64*	V64	сечового міхура
F5	GB40*	VB40	жовчного міхура
F6	ST42*	E42	шлунку

Виміряні в репрезентативних точках значення електропровідності заносяться в спеціальну R-картку, далі визначається середнє арифметичне значень електропровідності у всіх репрезентативних точках (24 значення, по 12 ліворуч і праворуч) і будується так званий «коридор норми» (див. рис. 2). За І. Накатані, меридіани, для яких значення електропровідності відповідної репрезентативною точки виявляються на графіку вище (червоним кольором) коридору норми, перебувають у стані «енергетичного надлишку», а меридіани, для яких значення електропровідності відповідної репрезентативною точки виявляються на графіку нижче коридору норми (синім кольором), знаходяться в стані «енергетичного нестачі». Слід підкреслити, що при визначенні стану меридіанів використовуються не абсолютні значення електричної провідності шкіри репрезентативних точок, а їх розташування щодо коридору норми (зеленим кольором) на R-картці [10].

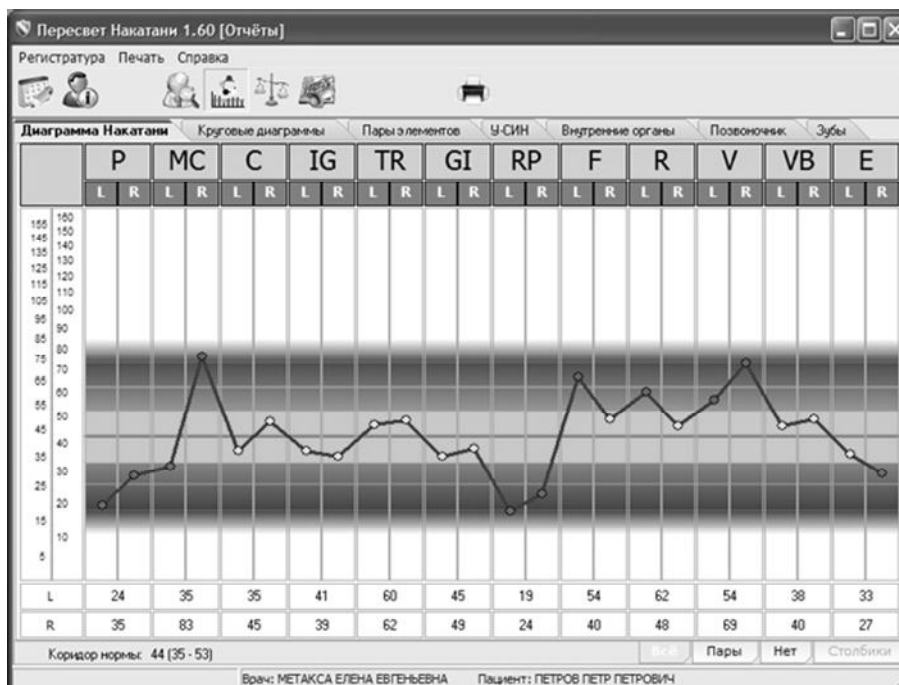


Рис. 2 Представлення результатів у вигляді R-картки

Багаторічні дослідження, показали, що на результати одноразового обстеження дуже часто впливає цілий ряд чинників, не пов'язаних із стійкими відхиленнями в меридіанах. До основних чинників такого роду можна віднести:

- 1) психоемоційний стан пацієнта на момент проведення дослідження;
- 2) порушення режиму харчування і сну напередодні проведення діагностики;
- 3) надмірні фізичні навантаження перед обстеженням;
- 4) прийом лікарських препаратів та ін.

Для підвищення якості діагностики та усунення помилок, викликаних випадковими факторами, рекомендується проводити діагностику способом повторних електропунктурних вимірювань: протягом 3-х днів, один раз на день, бажано в один і той же час.

Запропонована І. Накатані R-картка безсумнівно зручна при ручній обробці даних дослідження. Проте комп'ютерна реалізація методу І. Накатані вимагає математично точного опису параметрів R-картки з метою отримання порівняних результатів, при реалізації на різних апаратно-програмних комп'ютерних комплексах. В основу параметрів R-картки покладені формули, запропоновані В.М. Кімом (1998). Загальна апроксимуюча формула для кожного з меридіанів має вигляд [8]:

$$R_j = 0,866 \cdot H_{\max} \cdot L_n \cdot (1 \cdot I_j / b_j)$$

де R_j – висота точки, відкладається на шкалою R-картки для j -того меридіана; H_{\max} – максимальна висота шкали R-картки (мм); I_j – вимірне значення струму для j -того меридіана (мкА); b_j – коефіцієнт для j -того меридіана

представлений в таблиці 3, при цьому обрахунок ведеться по дванадцятьом меридіанам.

Таблиця 3

Коефіцієнти апроксимації для побудови R-картки

Меридиан	H1LU	H2PC	H3HT	H4SI	H5TE	H6LI	F1SP	F2LR	F3KI	F4BL	F5GB	F6ST
b_j	81	69	81	94	94	76	60	76	70	60	67	89

Апроксимуюча формула для середнього значення має такий самий вигляд, як і для окремого меридіана, коефіцієнт $b_{cp} = 76$.

Величина коридору норми розраховується за формулою:

$$DP = 0,133 \cdot H_{\max}$$

На відміну від попередньо описаного методу електроакупунктурної діагностики метод Накатані дає досить швидкі та достатньо точні результати при тому, що час діагностики займає близько 15-25 хвилин. При цьому точність ідентифікації хвороби варіює в діапазоні від 63,9% до 74,3 в залежності від прояву та стадії хвороби.

3.1.1.4 Ауromетрія

Людина є біоенергоінформаційною системою, яка складається з фізичного тіла та біополя яке знаходиться в серединні першого. Та частина біополя, яка знаходиться за межами матеріального тіла, називають аурую. Аура складається з тонких тіл: ефірного, ментального, астрального та інших (у різних джерелах їх називають по різному). Зазвичай кажуть, що тонких тіл – сім, а розвинені тільки три. У біополі знаходяться енергетичні центри – чакри, які перебувають у постійному обертанні і вібрації, а також енергетичні канали.

У 1939 році російські вчені, подружжя С.Д. і В.Х. Кірліан виявили світіння власних пальців, розміщених в електричному полі високої напруженості, що отримало назву «ефект Кірліан». У 70-80-х роках 20 століття сотні ентузіастів у всьому світі займаються кірліановскою фотографією, метод викликає великі надії і великі суперечки. У 1996 році в Санкт-Петербурзі російським академіком К. Г. Коротковим створено новий науковий підхід, заснований на сучасній електроніці, цифровий відеотехніці і кількісній комп'ютерній обробці даних – метод газорозрядної візуалізації (ГРВ) (ауromетрія). Метод ГРВ дозволяє зафіксувати зображення вимушеного світіння тел як живої, так і неживої природи, що знаходяться в полі високої напруги. ГРВ-грама є своєрідним енергоінформаційним «візерунком» кожного природного тіла.

Біоінформаційна адаптометрія (ауromетрія) – це комп'ютерний метод діагностики чакр і біоенергетичного поля людини, яка дозволяє вловити невидимі оку процеси на енергоінформаційному рівні, які довго перебуваючи в полі (аури) людини починають впливати на життєдіяльність людини, на здоров'я, взаємовідносини, кар'єру, бізнес, сім'ю, і багато інших сфер життєдіяльності людини. Відмітна особливість методу діагностики полягає в тому, що він виявляє

найслабшу ланку в енергоінформаційному полі (аури) людини і вказує на осередок патології (див. додаток Б). У плані здоров'я метод дає можливість виявити порушення у задовгу організмі до того, як з'являться перші клінічні симптоми захворювання [12].

За допомогою даних, отриманих після вимірювання енергетичного опору БАТ на пальцях рук і ніг, де знаходяться початкові і кінцеві точки всіх меридіанів залежно від напрямку течії енергії, можна відтворити енергоінформаційних голографічну матрицю (ауру). Такий підхід називається «методом енергоінформаційної адаптометрії» розроблений академіками Александром и Ольгою Влаховими [12].

Також, уявлення про меридіани та таблиці Мандела (секторальна діагностика світіння кожного пальця руки) використовуються для інтерпретації даних ГРВ-грам. Цей метод дозволяє з точністю до 95-97% діагностувати будь-які порушення циркуляції енергії в організмі людини на ранніх стадіях захворювання (за 5-7 років до появи хвороби у фізичному тілі), а також патологічні зміни в організмі на даний момент.

Даний метод діагностики, звичайно, з перебільшенням можна назвати діагностикою біополя. На даний момент не існує приладу для фіксації біоенергетичного поля людини. Цей метод показує всього лише коронне світіння навколо тіла. Тим не менше, оскільки методу вже більше 20 років, він досить добре розроблений, апробований та інформативний.

3.1.2 Апаратне та програмне забезпечення для біорезонансної діагностики

Сучасні апаратно-програмне забезпечення для біорезонансної діагностики має ряд модифікацій та особливостей пов'язаних їх функціональними можливостями. Апаратне забезпечення в більшості випадків складається з безпосередньо приладу для зняття показників, ідентифікації патології та генерації електричного сигналу для проведення терапії.

Також, не менш важливим елементом є електроди, за допомогою яких здійснюється отримання та передача сигналів. Для проведення електроакупної діагностики по методу Фоля чи Накатані використовують два види електродів: пасивні та активні (щуп). За допомогою останнього здійснюється запит до біологічно активних точок. Зазвичай щуп виготовлений з латуні, трубчастої форми з металевим наконечником сферичної форми. При сегментарній діагностиці використовуються плоскі електроди для ніг, циліндричні для рук та у формі пластин налобні. В апаратно-програмних комплексах, які дозволяють проводити діагностику по декільком методикам, використовуються електроди описаних вище типів.

Сучасні електроакупунктурні системи активно використовують комп'ютерну обчислювальну техніку (персональні комп'ютери – ПК) для представлення результатів діагностики та проведення контролю за нею. Для забезпечення обміну інформацією між ПК та приладом використовується з'єднання через COM та USB порти.

Біорезонансні діагностичні системи повинні генерувати напругу у досить малому діапазоні від 1,2 до 1,5 V та струм до 20 мкА (метод Фоля), та

забезпечувати терапевтичну дію імпульсною напругою в частотному діапазоні від 0,8 до 10 Гц [5].

На рис. 3 представлена принципова схема приладу для методу Фоля. Схема складається з недорогого операційного підсилювача, двох потенціометрів, стрілочного вимірювального приладу, двополюсного вимикача та п'яти постійних резисторів. Для живлення застосовані дві батареї формату ААА.

Принципи роботи схеми розглянемо щодо мінусового провідника джерела напруги та пасивного електроду. При включенні живлення вихідна напруга ОП, включеного за схемою інвертується повторювача, встановлюється рівній напрузі на виході 3, приєднаний через резистор R3 до плюсової виходу з джерела струму 2, і дорівнює напрузі 1,5 В, плюс напруга зсуву підсилювача, що компенсується потенціометром P1. При рівності напруг батареї 2 і виходу 6, попереднього збалансованого потенціометром P1 підсилювача, струму через вимірювальну ланцюг R4, P2 і вимірювальний прилад немає, і стрілка мікроамперметра встановлюється на нульове значення. Підключивши активний та пасивний електроди за схемою вхід R1 з від'ємним виходом батареї 2, через резистор R1 потече струм. ОП перейде в режим інвертування підсилювача з коефіцієнтом передачі, що дорівнює відношенню опорів R2/R1. Вихідна напруга ОП вимірюється стрілочним мікроамперметром, утримуючи струмом через резистор зворотного зв'язку R2 рівність вхідної напруги на входах 2 і 3, компенсує поточний струм через резистор R1. Максимальний розмах вихідної напруги ОП в 1В визначається номіналом резистора R2. Значення показань стрілочного приладу встановлюється потенціометром P2 на кінцеву позначку шкали «100» [5].

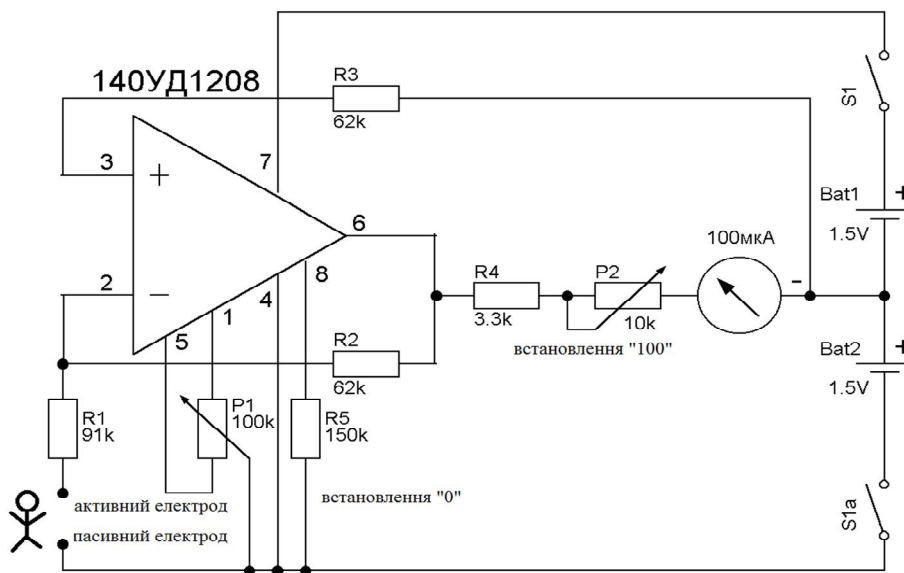


Рис. 3 Принципова схема приладу для діагностики по методу Фоля

Максимальна величина струму через вимірювальну ланцюг і електроди відповідає самим м'яким стандартам 16 мкА при замкнутих електродах і 1,55 В при розімкнутих електродах, визначається номіналом резистора R1 і напругою батареї 2. Номінал резистора R1, позначений на схемі, найближчий з десятивідсоткового ряду до необхідного в 95 кОм.

Інші системи мають подібну будову проте додатково можуть містити АЦП для роботи з комп'ютером, мікроконтролери та карти пам'яті. Принципова схема приладів та числові показники її елементів залежить від функціональної спрямованості обладнання.

На сьогоднішній день існує ціла низька апаратно-програмного забезпечення, яке за функціональною спрямованістю можна умовно поділити на спеціалізовані (реалізована одна методика) та комплексні (які дозволяють проводити діагностику за різними методами). До першої групи належать: «Мой домашній доктор», «МОРА», «ОБЕРОН», «Пересвет Фоль» [10], «Пересвет Накатани» [10] та інші. До другої групи відносять: «Лидомед-БИО» [9], «АРМ-ПЕРЕСВЕТ» [10], «АТМ Express» [12] та інші. Більшість з них дозволяють проводити біорезонансну терапію, підбір медикаментів, харчовий продуктів та інше.

В рамках даної роботи будуть розглянуті найбільш типові з медичних діагностичних систем на основі енергетичних процесів в організмі людини.

Першим з таких систем буде розглянуто програмний комплекс «АТМ Express» фірми Altimed. Даний програмно-апаратний комплекс складається з приладу АТМ який дозволяє дати оцінку загального стану організму, визначити реактивність і адаптивні можливості, тип вегетативної нервової системи, екзогенних, ендогенних, радіаційних навантажень, наявність або відсутність глистної інвазії, грибкового, бактеріального, вірусного обтяжень, пошкодження мікробіологічного циклу в кишечнику, крові, лімфі, стан центральної нервової системи, імунної, лімфатичної, ендокринної систем та хребта. Автоматично визначає органи-мішені, на які необхідно звернути увагу. На підставі інтегрального аналізу тридцяти показників, метод дає можливість прогнозувати ризик виникнення захворювань, здійснити їх ранню діагностику, оцінити умови прогресування патології. Даний пристрій має наступні можливості:

- 1) управління функціями приладу за допомогою комп'ютера;
- 2) можливість багаторазового зняття інформації, не вводячи похибки в контур система-пацієнт;
- 3) автоматичне калібрування протягом кількох секунд;
- 4) перевірка цілісності ланцюгів і контакту електродів з пацієнтом;
- 5) інтегрована система автоматичного контролю та знімання інформації [11].

Для представлення результатів діагностики та здійснення контролю за проведенням тестування та лікування пацієнта, використовується програмний комплекс «АТМ Express». Який оснований на методі сегментарної діагностики та протягом 30 секунд дозволяє отримати інформацію про всі органи та системи, їх фізіологічний стан і патологічні зміни у них, завдяки величезній селекторної базі (понад 35000 тестових резонансних частот) програми провести тестування на наявність широкого спектру збудників і патологічних станів, визначити індивідуальну схему терапії.

Терапевтичний комплекс складається із загальних і вибіркового програм біорезонансної терапії (БРТ). Останні, у свою чергу, діляться на меридіанні, системні та нозологічні програми. Широкий вибір програм БРТ дозволяє скласти максимально ефективну та комплексну схему терапії з урахуванням індивідуальних особливостей пацієнта.

Також «АТМ Express» дозволяє проводити діагностику за методом Фоля та використовувати вегетативно-резонансний тест. Завдяки великій селекторній базі програмно-апаратний комплекс дозволяє якісно підібрати медикаментозне лікування та індивідуальне харчування серед 800 найменувань продуктів. Що дозволяє проводити якісне та швидке лікування пацієнтів навіть на ранніх стадіях, коли ще немає клінічних проявів хвороби.

Наступним апаратно програмним комплексом є «АРМ-ПЕРЕСВЕТ» [10] який дозволяє проводити діагностику та лікування за наступними методами:

- 1) метод Р. Фолля;
- 2) аурикулярная діагностика і терапія;
- 3) діагностика за методом Накатані;
- 4) гомеопатія (реперторіум Дж. Т. Кента);
- 5) магнітоінфракрасная лазерна / квантова терапія;
- 6) електростимуляційна рефлексотерапія;
- 7) фітотерапія, натуро терапія;
- 8) експертний метод оцінки болю.

Програма «АРМ-ПЕРЕСВЕТ» дозволяє:

- 1) проводити панорамну комп'ютерну діагностику пацієнта і визначати захворювання на ранній стадії, коли вони ще не виявляються загальноприйнятими методами;
- 2) проводити корекцію функціонального стану організму індивідуально підібраними терапевтичними методиками;
- 3) ефективно контролювати і коректувати хід лікування;
- 4) виявляти вплив ліків, препаратів, побутової хімії, продуктів харчування, напоїв, косметики, металів та інших матеріалів на організм людини без введення їх всередину;
- 5) визначати ефективність фізіотерапевтичних процедур і будь-яких інших видів впливу на людину.

Багаторічна клінічна практика використання аурикулярної медицини при різних захворюваннях і особливих фізіологічних станах довела її високу ефективність і надійність [10].

Метод багатofакторної аурикулярної діагностики (МФАД) відрізняється комплексним аналізом електропровідності, больової чутливості і морфологічних трансформацій в аурикулярних точках (див рис. 4). Дана методологія була розроблена представниками наукової школи професора Р.А. Дуріяна на основі результатів фундаментальних і прикладних досліджень, а також оригінальних робіт провідних фахівців - В.І. Квірчішвілі, 1966; Д.М. Табеева, Л.М. Клименко, 1976; В.Г. Вогралік, М.В. Вогралік, 1978; Ф.Г. Портнов, 1980; Є.С. Вельховер, В.Г. Нікіфоров, 1984; Я.С. Песиків, С.Я. Рибалко, 1990; А.А. Михайлова, 1992; Ю.А. Ткаченко, 2000; Р. Nogier, 1956; Н.С. Lu, 1975; Н. Kropej, 1976.

Модуль аурикулярної рефлексодіагностики дозволяє проводити експрес-та поглиблену діагностику. Для проведення експрес-обстеження розроблений маршрут, який дозволяє отримати загальну інформацію про стан здоров'я пацієнта, використовуючи найменшу кількість аурикулярних точок. Крім скрінінгового дослідження в модулі «Пересвіт Аура» можна вибрати маршрути повного обстеження, за всіма або окремими системами органів (усього 11), зонами вушної раковини або в іншому довільному порядку.

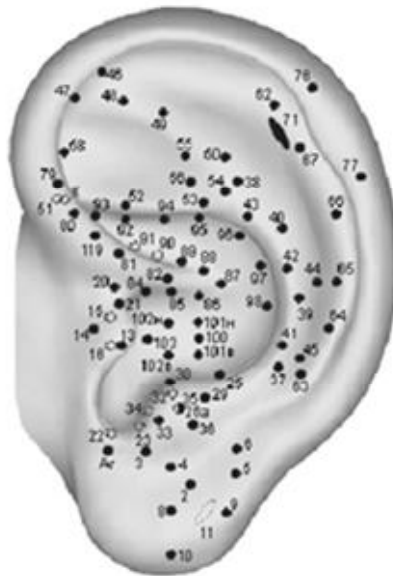


Рис. 4 Аурикулярні точки

Процес аурикулярної діагностики оснований на наступних чинниках:

- 1) оцінка факторів електроаномальності;
- 2) оцінка больової чутливості;
- 3) реєстрація візуальних змін в аурикулярних зонах.

Результати діагностики представляються у табличному вигляді, а також даний програмний продукт забезпечує графічне представлення внутрішніх органів та зон хребта, зафарбованих різними кольорами, відповідно до рівня загальної аномальності точок (ОА), характеру перебігу патологічного процесу (див. рис. 5).

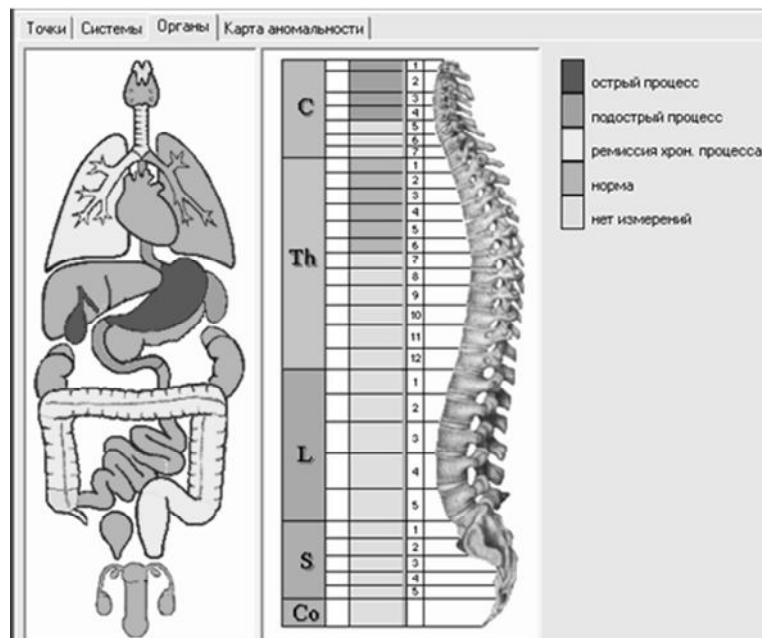


Рис.5 Графічне представлення результатів діагностики

Різноманітне апаратно-програмне забезпечення біорезонансної діагностики і терапії має ряд спільних ознак, оскільки працює на основі вимірювань енергетичного потенціалу в біологічно активних точках. Проаналізувавши існуючі системи було виявлено ряд перспективних напрямків розвитку біорезонансної діагностики, пов'язаних з використанням мобільних технологій, комп'ютерних мереж та методів прогнозування впливу лікування та розвитку патології.

3.1.3 Контроль і прогнозування змін станів пацієнта під час діагностування та біорезонансної терапії

Контроль за змінами стану пацієнта має досить важливе значення, оскільки характеризує не лише ефективність обраного плану лікування, а й дозволяє виявити критичні моменти та встановити залежності для усунення проблем в подальшому. При формуванні методів прогнозування треба розробити модель, яка характеризує процес зміни параметрів системи органів людини під впливом біорезонансних сигналів та навколишнього середовища. При формуванні моделей досить складним є визначення параметрів та їх вплив на зміну стану пацієнта, оскільки людський організм є складною системою, яка містить безліч показників та залежностей.

На даний момент при прогнозуванні розвитку патології використовують результати кількох сеансів діагностики, порівнюють їх значення та методи впливу і формулюють наступні рекомендації, що пов'язані з формуванням подальшого плану лікування.

Для вирішення поставлених задач в системах реального часу необхідним є проведення порівняльної характеристики застосовності різних методів прогнозування та визначення оптимального з них для використання в мобільних діагностичних системах.

3.2. Розробка вимог до програмного забезпечення для біорезонансної діагностики і терапії

3.2.1 Розробка загальних вимог до створюваного ПЗ

Проаналізувавши предметну область та дослідивши існуючі апаратно-програмні рішення біорезонансної діагностики було сформовано ряд вимог та напрямків розвитку галузі, на основі яких буде створено мобільний апаратно-програмний комплекс біорезонансної діагностики та терапії. Створюваний комплекс повинен забезпечувати:

- 1) проведення діагностики за наступними методами:
 - a) метод Фоля;
 - b) метод сегментарної діагностики;
 - c) Vega-тест;
 - d) метод Накатані;

- e) Ауromетрія;
 - f) біорезонансна терапія (пасивна та активна);
 - g) спосіб підбору харчових продуктів.
- 2) проведення періодичного контролю за перебігом лікування та прогнозування зміни стану пацієнта.
 - 3) використання мобільного апаратного та програмного забезпечення для контролю, аналізу та представлення результатів;
 - 4) дистанційну діагностику та терапію використовуючи наступні типи мереж:
 - a) Wifi;
 - b) Bluetooth;
 - c) Internet;
 - d) локальні мережі.
 - 5) усунення завад та артефактів викликаних внутрішніми та зовнішніми завадами;
 - 6) реалізація протоколів передачі даних, що забезпечує захист медичної інформації;
 - 7) підтримка транзакцій при використанні мережевого режиму роботи;
 - 8) можливість візуалізації результатів діагностики у вигляді схем, моделей, таблиць та інше;
 - 9) збереження хронології проведення діагностики та лікування з можливістю їх порівняння.
 - 10) можливість напівавтоматичного формування результатів діагностики;
 - 11) створення звітів з можливістю їх друку та передачі у середовище Microsoft Word та інші редактори.

3.2.2 Мобільні технології в енергетичній медицині, перспективи та критерії використання

Використання мобільних засобів набуло досить широкого розповсюдження та ефективного використання у багатьох сферах людського життя. Перевагою цих пристроїв є доступність, компактність, легкість використання, проте застосування мобільних технологій зобов'язує розробника враховувати обмеженні ресурсів КПК або інших мобільних пристроїв при створенні власних програмних засобів. Основними перевагами їх використання в медичній галузі являється їхня придатність для дистанційного моніторингу стану хворих та контролю навантаження під час тренувань у спортсменів, властивий їм високий рівень автономності роботи. На сьогодні серед великого різноманіття програмного забезпечення для мобільних комп'ютерних засобів програми медичного спрямування практично відсутні, що частково зумовлено складністю математичного та алгоритмічного забезпечення розв'язання медичних діагностичних задач та необхідністю використання зовнішнього апаратного устаткування.

На сьогоднішній день можна виділити наступні напрямки використання мобільних засобів в медицині:

- 1) контроль прийому медикаментів;
- 2) забезпечення моніторингу життєвих функцій пацієнта;

- 3) автоматизація процесу збору аналізів;
- 4) ведення лікарняних оглядів використовуючи електронні історії хвороби;
- 5) дистанційний ЕКГ моніторинг.

Не зважаючи на досить розвинену методологічну базу діагностики на основі біологічно активних точок та різноманіття апаратно-програмних засобів на сьогодні не існує мобільних діагностичних систем в області медицини. Частково це можна пояснити впливом електромагнітних полів на результати діагностики та обмеженими обчислювальними можливостями, а також особливостями взаємодії мобільних засобів та приладу. Проте дана галузь недостатньо досліджена і вимагає більш детального аналізу проблеми, пов'язаної з вибором критеріїв оцінки мобільного апаратного забезпечення та каналів зв'язку з приладом і сервером для реалізації можливості дистанційної діагностики та терапії. В якості критеріїв оцінки виступає вплив параметрів мобільного апаратного забезпечення на ефективність роботи системи її обчислювальні та якісні показники.

3.2.3 Дистанційна біорезонансна діагностика та терапія

Даний напрямок не новий та знайшов своє широке використання в різних галузях медицини. В багатьох країнах світу створюються регіональні та національні медичні системи, основані на єдиному програмному забезпеченні та стандартах апаратного забезпечення. Так у Фінляндії була створена регіональна та об'єднана медичні інформаційні системи HUS та HIS відповідно.

На даний момент існує декілька пробних систем використання дистанційної біорезонансної терапії, проте даний напрямок потребує більш глибокого дослідження пов'язаного з вибором типу мережу (Wifi, Bluetooth, Internet, локальні мережі), її топології та протоколів передачі даних (використання стандартного медичного протоколу HL7, або розробка спеціалізованого протоколу), а також з вибором оптимальних методів шифрування даних, з метою їх захисту від несанкціонованого втручання. Використання клієнт-серверної архітектури дає ряд можливостей:

- 1) забезпечення оновлення бази знань;
- 2) можливість збереження електронних історій хвороб на сервері з метою подальшого аналізу розвитку патології та оцінки ефективності методу лікування;
- 3) дистанційний контроль за проведенням діагностики та лікування;
- 4) моніторинг за станом кількох пацієнтів одночасно.

При використанні дистанційної біорезонансної діагностики існує можливість виникнення помилок пов'язаних з відсутністю навичок застосування акупунктурних методів, пов'язаних з методом Фоля та Накатані. Тому перспективним є використання дистанційної діагностики на основі Vega-тесту та сегментарної діагностики, а також активної та пасивної БРТ.



Висновки

В даній роботі проведений огляд методів діагностики стану пацієнтів на основі біологічно активних точок. Також, представлені апаратні характеристики для проведення біорезонансної діагностики та терапії. В процесі розгляду існуючих програмних комплексів було встановлено ряд напрямків, які потребують подальшого вивчення та дослідження, першим з яких є використання мобільних засобів для обробки, представлення та передачі медичної інформації.

Наступним напрямком розвитку даної галузі є можливість дистанційної діагностики і лікування пацієнта, а також, можливість on-line підтримки пацієнтів та лікарів. Перспективним напрямком є створення єдиного простору для збереження історії хвороби багатьох пацієнтів, що дозволить на основі отриманих даних створювати експертні системи для формування плану лікування та контролю його перебігу. Проте, при використанні мережевих систем проблемою являється вибір типу мережі та протоколу передачі даних для забезпечення стабільності роботи і захисту особистої інформації клієнта. Розвиток даних напрямків та їх впровадження дозволить створити медичні системи нового покоління, що значно покращить якість діагностики, швидкодію та надійність створених програмно-апаратних комплексів.

Список використаних джерел

- 1) Синди Дейл. Чакри. – СПб.: София, 2006. – 256 с.
- 2) Аванесова Е.Г., Аванесова Т.С., Готовский М.Ю., Бочаров Д.Г. Применение вегетативного резонансного теста "ИМЕДИС-ТЕСТ" в клинической практике: Руководство. – М.: Имедис, 2007. – 164 с.
- 3) Г.А. Юсупов. Энергоинформационная медицина. – Изд. дом «Московские новости», 2000. – 335с.
- 4) Я. В. Пишель, М.И. Шапиро. Анатомо-клинический атлас рефлексотерапии. – Харьков: ОКО, 1995.
- 5) А.В. Самохин, Ю. В. Готовский. Электропунктурная диагностика и терапия по методу Р. Фолля. 5-е изд., стереотипное. – М.:Имедис, 2006. – 528 с.
- 6) Ф. Г. Портнов. Электропунктурная рефлексотерапия. – Рига: Зинатне, 1987. – 325 с.
- 7) Шиммель Х.В. Фотон-резонансный тест – инновационное дополнение ЭАП. Материалы 8-й международной конференции «Теоретические и клинические аспекты применения биорезонансной и мультирезонансной терапии. – М.: Имедис, 2002.
- 8) Бойцов И.В. Электропунктурная диагностика по «риодораку». – Витебск, 1996. – 192 с.
- 9) Лидомед-Био [Електронний документ]. Режим доступу: <http://10001.lidomedbio.com/product> . Перевірено: 09.09.2011.
- 10) Комплекс АРМ-ПЕРЕСВЕТ [Електронний документ]. Режим доступу: <http://www.peresvetmed.ru/prsvtabout.html> . Перевірено: 09.09.2011.
- 11) АТМ-Expres [Електронний документ]. Режим доступу: <http://www.altimed.net> . Перевірено: 09.09.2011.
- 12) Влахов А. Диагностика, аураграфия и информационно-резонансная коррекция. // Стиль жизни. – 2000. – № 3.

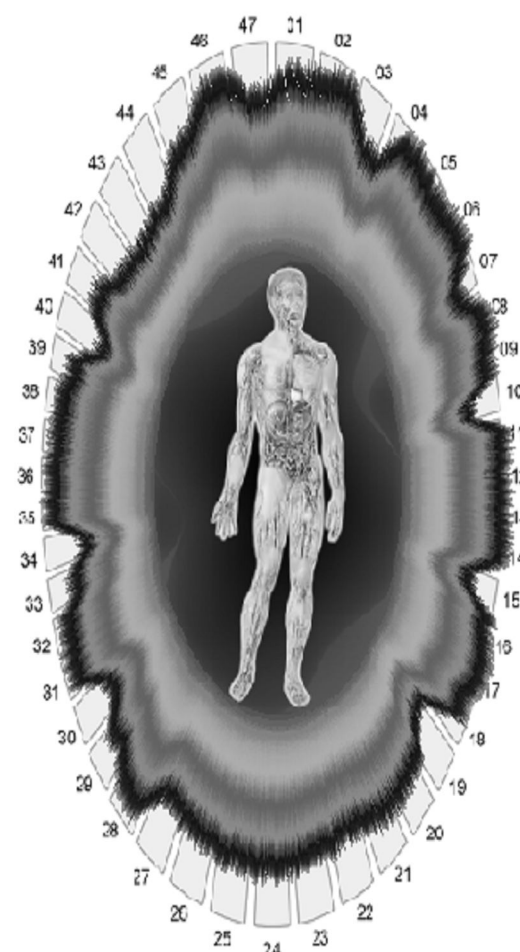
Шкали значень і діаграми для сегментарної діагностики

№	Відповідність органу та системі	Діапазон показників норми
1	Мозковий кровообіг праворуч	40 - 80
2	Мозковий кровообіг зліва	40 - 80
3	Венозна система (праві відділи серця)	75 – 85
4	Артеріальна система (ліві відділи серця)	75 - 85
5	Органи зору і слуху праворуч	70 - 80
6	Органи зору і слуху ліворуч	70 - 80
7	Гайморові і фронтальні порожнини праворуч	70 - 80
8	Гайморові і фронтальні порожнини зліва	70-80
9	Гіпоталамус	40 - 80
10	Гіпофіз	40 - 80
11	Тимус	70 - 80
12	Щитовидна залоза права частка	70 - 85
13	Щитовидна залоза ліва частка	70 - 85
14	Наднирники праворуч	72 - 88
15	Наднирники ліворуч	72 - 88
16	Молочні залози праворуч	70 - 83
17	Молочні залози зліва	70 - 83
18	Бронхи праворуч	70 - 85
19	Бронхи зліва	70 - 85
20	Легені праворуч	70 - 85
21	Легені зліва	70 - 85
22	Гортань, трахея	75 - 85
23	Лімфатичні вузли середостіння	70 - 80
24	Лімфатичні вузли шиї	70 - 80
25	Мигдалини глоткового кільця праворуч	70 - 80
26	Мигдалини глоткового кільця зліва	70 - 80
27	Селезінка	75 - 85
28	Печінка	72 - 88
29	Жовчний міхур	72 - 82
30	Шлунок	72 - 82
31	Дванадцятипала кишка	78 - 88
32	Горизонтальний відділ товстого кишечника	72 - 82
33	Тонкий кишечник	75 - 85
34	Висхідний відділ товстого кишечника, апендикс	75 - 85
35	Низхідного відділу товстого кишечника, пряма кишка	72 - 85
36	Підшлункова залоза	72 - 82
37	Глотка, стравохід	70 - 80
38	Матка (Ж) простата (М)	72 - 82
39	Сечовий міхур	72 - 82
40	Яєчник праворуч (Ж) яєчка праворуч (М)	72 - 88
41	Яєчник ліворуч (Ж) яєчка зліва (М)	72 - 88
42	Брунька, сечовід праворуч	78 - 88
43	Брунька, сечовід зліва	78 - 88
44	Суглоби рук справа	78 - 85
45	Суглоби рук зліва	78 - 85
46	Суглоби ніг праворуч	78 - 88
47	Суглоби ніг зліва	78 - 88

Ауромерія

aur.info

Состояние ауры



09.10.2009 14:28:45

- 01 ● -5 Левое полушарие головного мозга, и его сосуды.
- 02 ● 0 Ствол головного мозга, мозжечок левое полушарие.
- 03 ● -30 Орган зрения и слуха слева
- 04 ● 0 Слизистая носа и придаточных пазух слева
- 05 ● -8
- 06 ● -2 Миндалины и лимфопитательное кольцо слева
- 07 ● -22 Лимфоотток от органов головы и шеи
- 08 ● 0 Пищевод
- 09 ● -9 Грудной лимфатический проток
- 10 ● -38 Щитовидная железа слева
- 11 ● 0 Артериальная система, левое сердце
- 12 ● -4 Бронхи слева
- 13 ● -2 Легкое слева
- 14 ● 0 Суставы рук и плечевого пояса
- 15 ● -36 Селезенка
- 16 ● -8 Желчный пузырь и желчевыводящие пути
- 17 ● 0 Желудок
- 18 ● -12 Тонкий кишечник
- 19 ● -50 Нисходящий отдел толстого кишечника, прямая кишка
- 20 ● -28 Надпочечник левый
- 21 ● -26 Почка левая
- 22 ● -33 Мочевой пузырь
- 23 ● -30 Яички слева
- 24 ● -26 Суставы ноги тазового пояса с лева
- 25 ● -26 Суставы ноги тазового пояса с права
- 26 ● -28 Яички справа
- 27 ● -32 Простата
- 28 ● -2 Почка правая
- 29 ● -18 Надпочечник правый
- 30 ● +33 Восходящий отдел толстого кишечника
- 31 ● 0 Горизонтальный отдел толстого кишечника
- 32 ● +4 12 перстная кишка
- 33 ● -22 Печень
- 34 ● -37 Поджелудочная железа
- 35 ● 0 Суставы рук и плечевого пояса
- 36 ● -2 Легкое справа
- 37 ● -5 Бронхи справа
- 38 ● -7 Венозная система, правое сердце
- 39 ● +18 Щитовидная железа справа
- 40 ● -42 Тимус
- 41 ● -24 Гортань
- 42 ● -41 Миндалины и лимфопитательное кольцо справа
- 43 ● -46
- 44 ● -40 Слизистая носа и придаточных пазух справа
- 45 ● -22 Орган зрения и слуха справа
- 46 ● 0 Ствол головного мозга, мозжечок правое полушарие
- 47 ● -18 Правое полушарие головного мозга, и его сосуды

Наукове видання

Парнюк Анна Миколаївна

Волошин Миколай Володимирович

Єремєєв Богдан Миколайович

**МОДЕЛЮВАННЯ
ПАРАЛЕЛЬНИХ ПРОЦЕСІВ
В ІНФОРМАТИЦІ ТА МЕДИЦИНІ
МОНОГРАФІЯ**

*за науковою редакцією
проф. Кузьмука В.В. та доц. Супруненко О.О.*

*Комп'ютерна верстка:
А.М. Парнюк, В.В. Волошин, Б.М. Єремєєв*

Підписано до друку 23.12.2011. Формат 60?84/16.
Ум. друк. арк. 6,6. Тираж 100 прим. Зам. № 4110.

Видавець і виготовлювач
Черкаський національний університет імені Богдана Хмельницького
Адреса: 18000, м.Черкаси, бул.Шевченка, 81, кімн. 117,
Тел. (0472) 37-13-16, факс (0472) 37-22-33,
e-mail: vydav@cdu.edu.ua, <http://www.cdu.edu.ua>
Свідоцтво про внесення до державного реєстру
суб'єктів видавничої справи ДК №3427 від 17.03.2009 р.