

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»

Фізико-математичний факультет  
(повна назва інституту/факультету)

Кафедра загальної фізики та фізики твердого тіла  
(повна назва кафедри)

«На правах рукопису»  
УДК 004.4:681.586.67](043.3)

«До захисту допущено»

Завідувач кафедри  
\_\_\_\_\_ Котовський В. Й.  
(підпис) (ініціали, прізвище)

“\_\_\_” грудня 2018 р.

## Магістерська дисертація

зі спеціальності (спеціалізації) 104 фізика та астрономія  
(код і назва спеціальності)

на тему: Розробка та впровадження удосконаленого програмного забезпечення до термографа української розробки

Виконав: студент 2 курсу магістерського рівня, групи

ОФ-71мп  
(шифр групи)

Шевченко Веніамін Сергійович  
(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник д.техн.н., проф. Котовський В.Й.  
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2018 року

## **ЗАВДАННЯ**

на магістерську роботу студента

Шевченка Веніаміна Сергійовича

**1. Тема роботи:** «Розробка та впровадження удосконаленого програмного забезпечення до термографа української розробки».

Керівник роботи: д.техн.н.,проф.Котовський В.Й

(Наукова ступінь, посада, ПІБ)

Затверджено наказом по університету від «02» листопада 2018р. №4064-с

**2. Строк подання студентом роботи** до 12 грудня 2018 р.

**3. Вихідні дані до роботи:** програмне забезпечення до термографа ТВС-300, середовище програмування Borland Delphi 7.

**4. Зміст розрахунково-пояснювальної записки** (перелік завдань, які потрібно розробити):

- Розробити алгоритми:
  - прибирання шумів;
  - виділення окремих температурних шарів;
  - визначення площі об'єкта дослідження;
  - візуалізація термограми у вигляді 3D графіку.
- Програмно реалізувати розроблені алгоритми.
- Провести тестування роботи програми.

**5. Перелік графічного матеріалу:** 20 рисунків

**6. Дата видачі завдання:** 03.09.2018

### Календарний план

Назва етапів виконання магістерської дисертації	Строк виконання етапів роботи	Примітка
Перегляд літературних джерел, складання огляду літератури	03.09.18-17.09.18	
Аналіз існуючого програмного забезпечення	18.09.18-30.09.18	
Розробка нових алгоритмів для підвищення якості термограм	01.10.18-15.10.18	
Реалізація нових алгоритмів за допомогою мови програмування Delphi 7	16.10.18-25.10.18	
Пошук та виправлення логічних помилек в програмному коді	26.10.18-31.10.18	
Тестування програми	01.11.18-05.11.18	
Аналіз отриманого результату	06.11.18-10.11.18	
Оформлення дипломної роботи	11.11.18-24.11.18	
Попередній захист	12.12.18	
Корегування роботи відповідно до зауважень комісії	12.12.18-14.12.18	
Остаточний захист роботи	18.12.18	

Магістр \_\_\_\_\_  
(підпис)

Шевченко В. С.  
(ініціали, прізвище)

Керівник роботи \_\_\_\_\_  
(підпис)

Котовський В. Й.  
(ініціали, прізвище)

## Реферат

*Актуальність:* теплобачення, як перспективна галузь розвитку науки і техніки, особливо важливе місце займає в медицині при виявленні патологій на ранніх стадіях їх виникнення. Своєчасне діагностування захворювань та правильна постановка діагнозу дає можливість попередити та адекватно лікувати більшість хвороб зі зменшенням затрат на лікування.

Дана робота присвячена питанню розробки та вдосконалення програмного забезпечення з метою спрощення роботи з термографом та підвищення якості термографічних зображень.

*Постановка задачі:* аналіз термографічного зображення є важливим етапом при діагностуванні патологічних станів людини. Удосконалене програмне забезпечення надасть можливість прискорювати аналіз термограм, та попереджувати помилки оператора-дослідника.

*Шляхи вирішення проблеми:* розробка стійких алгоритмів та реалізація цих алгоритмів за допомогою програмного коду.

*Результати та висновки:*

- отримано термографічні зображення з видаленими фоновими шумами та пошкодженими пікселями;
- розроблено та реалізовано програмний код для знаходження екстремальних значень температур;
- спрощено отримання осцилографічних термозрізів в реальному значенні температур;
- розроблено спосіб визначення площі ділянки об'єкту досліджень;
- розроблено спосіб отримання тривимірного зображення з можливістю вимірювання температури по всій поверхні об'єкту досліджень, що значно розширює діагностичні можливості інтерпретації термографічних зображень.

*В роботі наведено:* використаної літератури - 29 , сторінок: - 108, рисунків- 20.

## Abstract

*Actuality:* thermovision, as a promising branch of development of science and technology, is especially important in medicine when pathologies are detected in the early stages of their occurrence. Timely diagnosis of diseases and correct diagnosis make it possible to prevent and adequately treat most diseases with a reduction in the cost of treatment.

This work is devoted to the development and improvement of software in order to simplify the work with the thermograph and improve the quality of thermographic images.

*Objectives of the study:* the analysis of the thermographic image is an important step in the diagnosis of human pathological conditions. Improved software will provide an opportunity to accelerate the analysis of thermograms, and to alert the operator of the researcher.

*Research methods:* the development of stable algorithms and the implementation of these algorithms using code.

*Results and their novelty:*

- received thermographic images with removed background noise and damaged pixels;
- the software code for finding extreme temperature values was developed and implemented;
- simplified obtaining of oscillographic thermal cuttings in the real value of temperatures;
- the method of determining the area of the site of the object of research is developed;
- a method for obtaining a three-dimensional image with the ability to measure temperature across the entire surface of the object of research is developed, which greatly expands the diagnostic capabilities of the interpretation of thermographic images.

*The paper presents:* used literature - 29, pages: - 108, drawings - 20.

## Зміст

Вступ.....	7
1. Основні фізичні закони, які лежать в основі отримання термографічних зображень .....	10
1.1 Закон Кірхгофа .....	10
1.2 Закон Стефана - Больцмана. Закон Віна. Формула Релея - Джінса.....	15
1.3 Формула Планка.....	19
2. Способи отримання термографічних зображень .....	23
2.1 Цифровий контактний термограф .....	23
2.2 Фоточутливі приймачі, які застосовуються в термографах .....	25
2.2.1 Болومتر .....	25
2.2.2 Приймачі ІЧ-зображення на основі термомеханічних нанорозмірних мембран .....	26
2.2.3 Фотоприймачі на основі теллуридів кадмію і ртуті .....	29
3. Способи перетворення інфрачервоного випромінювання в зображення на екрані монітора.....	30
3.1 Будова тепловізора.....	30
3.2 Опис діючої програми в термографі з охолоджуваною матрицею сенсорів .....	31
3.3 Удосконалення програмного забезпечення термографа ТВС-300.....	43
3.3.1 Розробка програми по видаленню фонових шумів .....	44
3.3.2 Розробка програми по виділенню окремих температурних шарів .....	47
3.3.3 Розробка програми по переведенню температури в реальні значення .....	48
3.3.4 Розробка програми для визначення площі ділянки об'єкту досліджень .....	49
3.3.5 Розробка програми для побудови 3D графіку.....	52
4. Розробка оновленого програмного забезпечення .....	53
4.1 Опис нової програми .....	53
4.2 Інструкції по використанню нових функцій програми.....	61
Висновки .....	64
Література .....	65
Додаток А .....	68

## Вступ

Інфрачервона (ІЧ) термографія (тепlobачення) є прикладним методом, який надає можливість вимірювати емісію інфрачервоних променів, випромінюваних кожним тілом, температура якого вище абсолютного нуля. На відміну від інших ІЧ методів, тепlobачення надає можливість «бачити» теплові промені, які випромінюють нагріті об'єкти за дуже короткий період часу. Перевагами ІЧ термографії в порівнянні з іншими методами є: висока швидкість визначення температури, висока чутливість сенсора, можливість вимірювати безконтактним способом, що не впливає на процес, а також можливість спостереження на одному термографічному знімку розподілу температури по всій поверхні об'єкту, будь-яких розмірів.

Останнім часом на ринку з'явилася велика кількість виробників сучасної термографічної техніки. Тепловізор (рівноважна назва термограф або ІЧ камера) робить знімок ІЧ випромінювання об'єкту досліджень або його конкретної ділянки. Спеціальне програмне забезпечення надає можливість перетворювати отриману інформацію у відповідну термограму, представляючи близькі до точних значення температури об'єкту досліджень або розподіл температур на відповідних ділянках. Розподіл температур по поверхні об'єкту дослідження відображується різними кольорами (псевдо-колір).

Термограми можна знімати безперервно та записувати у пам'ять термографа, а потім аналізувати

Камера дозволяє отримувати термограми в упорядкованих часових інтервалах та фіксувати певні відхилення, що попереджується певним сигналом. Це можуть бути відхилення від максимального і мінімального значення, середнього значення, і (або) показники температури в окремій точці, групі точок, тощо.

Оперативна пам'ять забезпечує запам'ятовування численних термограм, які за бажанням можна перенести на жорсткий диск комп'ютера.

Сучасні тепловізори додатково обладнані камерою видимого діапазону, що дозволяє автоматично зберігати одночасно і термограму, і реальний знімок

об'єкта. Функція накладання поєднує термограму і реальний знімок, та відображає поєднане зображення. Ця функція підвищує точність визначення розташування проблемної ділянки. Регулювання рівня прозорості задає інтенсивність компонентів реального та ІЧ знімка на суміщеному зображенні [16].

В останні роки спостерігається надзвичайно швидкий розвиток тепловізійної техніки та методів її застосування в медичній діагностиці [7-9,13-15]. Більшість причин, які обмежують застосування даного методу діагностики на сьогодні усунуті [5-8].

Сучасні матричні ІЧ-камери мають температурну чутливість порядку 0,007-0,015 °С при просторовій роздільній здатності 640 × 480. Краща чутливість матричних ІЧ-камер (в порівнянні з ІЧ камерами на основі одноелементних ІЧ-фотоприймачів) забезпечується внаслідок одночасної реєстрації ІЧ зображення в межах одного кадру всіма елементами матриці ІЧ-фотоприймачів. Це дозволяє істотно (а саме, в N×M раз, де N×M - розмір матриці фотоприймачів) збільшити час інтегрування ІЧ випромінювання в межах одного кадру.

У теперешній час також вирішено питання про співвідношення між термографічним зображенням на поверхні тіла біологічного об'єкту (БО) й анатомічними структурами, які розташовані на певній глибині від поверхні шкірного покриву. Встановлено, що важливу роль у формуванні теплової картини на поверхні тіла БО відіграє кровоплин, який забезпечує набагато кращу передачу тепла в порівнянні з класичною теплопровідністю гомогенних тканин [8]. Капіляри в даному випадку грають роль «теплових хвилеводів».

Практично зняті обмеження, пов'язані з високою вартістю тепловізійної техніки. З'явилися мініатюрні матричні ІЧ-камери, вартість яких порівняна з вартістю сучасних смартфонів, що працюють в спектральному діапазоні 8-12 мкм, що володіють температурною чутливістю порядку 0,03-0,04 °С при швидкості реєстрації 25 кадрів в секунду. Це цілком достатньо для діагностики більшості запальних і судинних патологій [7-8]. На основі таких ІЧ-камер можуть бути створені портативні (і навіть персональні) діагностичні системи, які

можуть знайти широке застосування в медичних закладах первинної ланки і машинах швидкої допомоги. Більш того, процедура реєстрації ІЧ-зображень може проводитися навіть в домашніх умовах з подальшим відправленням результатів вимірювань лікаря з використанням засобів телемедицини.

Доведено перспективність застосування методу ІЧ термографії в діагностиці запальних респіраторних і судинних захворювань, інтенсивно ведуться дослідження можливості ранньої діагностики онкологічних захворювань [2-4,10,12,13].

Термограма, розрахована на основі власного випромінювання тіла в ІЧ діапазоні довжин хвиль, на відміну від рентгенографії, несе інформацію про характер і інтенсивності фізіологічних процесів, що протікають в організмі, які можуть супроводжуватися аномальним теплоутворенням.

# 1. Основні фізичні закони, які лежать в основі отримання термографічних зображень

## 1.1 Закон Кірхгофа

Випромінювання світла, що відбувається внаслідок переходів атомів, молекул та інших атомних систем зі станів з більшою в стани з меншою енергією називають тепловим або температурним випромінюванням. Воно відрізняється від інших видів випромінювання (люмінесценції) тільки способом переходу випромінювальних систем у збуджений стан. В явищах тепловою випромінювання такий перехід здійснюється завдяки тепловому рухові томів і молекул.

В явищі теплового випромінювання особливо важливе значення має поняття так званого рівноважного випромінювання. Для визначення цього поняття розглянемо порожнину з нерухомими та непрозорими стінками температура яких підтримується сталою. Атоми та молекули стінок переходять у збуджений стан завдяки енергії теплового руху і при зворотних переходах у незбуджений стан випромінюють енергію. Променева енергія, падаючи на стінки порожнини, частково відбивається та частково поглинається. Внаслідок цього відбувається зміна напрямку поширення, спектрального складу, поляризації та інтенсивності випромінювання. Як наслідок всіх цих процесів у порожнині встановиться рівноважний стан, за якого величина енергії, що випромінюється стінками порожнини за одиницю часу, дорівнюватиме енергії, що поглинається цими ж стінками за одиницю часу. Через це стан випромінювання в порожнині залишається макроскопічно незмінним у часі. Таке випромінювання називають рівноважним.

Рівноважне випромінювання має такі властивості: густина променевої енергії, її розподіл у спектрі за довжинами хвиль (за частотами), напрям поширення, а також поляризація випромінювання не залежать від форми та речовини стінок порожнини. Ці властивості, подібно до стану газу в посудині, зумовлюються тільки температурою стінок порожнини. Рівноважне випромінювання однорідне: його густина однакова у всіх точках порожнини.

Воно ізотропне і неполяризоване, тобто всі напрями поширення випромінювання в порожнині рівноймовірні, а напрями векторів напруженості електричного поля  $\vec{E}$  та індукція магнітного поля  $\vec{B}$  у кожній точці порожнини змінюються з часом хаотично.

Оскільки випромінювання знаходиться в тепловій рівновазі зі стінками порожнини, то можна говорити не тільки про температуру стінок, а й про температуру самого випромінювання вважаючи ці температури однаковими.

Введемо деякі величини, які характеризують стан випромінювання у просторі, й застосовні не тільки для рівноважного, але і для довільного випромінювання.

Однією з таких величин є *густина енергії випромінювання*  $u$ , тобто кількість енергії випромінювання в одиниці об'єму. Густина енергії випромінювання можна виразити так:

$$u = \int_0^{\infty} u_{\nu} d\nu = \int_0^{\infty} u_{\lambda} d\lambda, \quad (1)$$

де величини  $u_{\nu} d\nu$  і  $u_{\lambda} d\lambda$  мають зміст об'ємної густини променевої енергії, що припадає на інтервал частот  $\nu, \nu + d\nu$  або інтервал довжин хвиль  $\lambda, \lambda + d\lambda$ . Коефіцієнт  $u_{\nu}$  і  $u_{\lambda}$  називаються спектральними густинами променевої енергії. Для випадку рівноважного випромінювання функції  $u_{\nu}$  і  $u_{\lambda}$  залежать тільки від частоти  $\nu$  (або довжини хвилі  $\lambda$ ) та від температури випромінювання  $T$  і не залежать від форми та матеріалу стінок порожнини. Для випадку, коли у порожнині вакуум,  $u_{\nu}$  буде універсальною функцією тільки  $\nu$  і  $T$ , а  $u_{\lambda}$  універсальною функцією тільки  $T$ . Встановлення функціональної залежності  $u_{\nu} = f(\nu, T)$  в явній формі є основним завданням теорії теплового випромінювання.

Перед тим, як аналізувати закони теплового випромінювання і поглинання тіл, введемо ще ряд необхідних понять.

*Випромінювальна здатність*  $E(T)$  чисельно дорівнює енергії, яку випромінює тіло при даній температурі з одиниці поверхні за одиницю часу в усьому інтервалі частот.

*Спектральна випромінювальна здатність тіла*  $E(\nu, T)$  чисельно дорівнює енергії, що випромінюється при заданій температурі одиницею поверхні за одиницю часу в інтервалі частот  $\nu, \nu + d\nu$  або інтервалі довжин хвиль  $\lambda, \lambda + d\lambda$ .

Повна (сумарна, інтегральна) випромінювальна здатність тіла визначається так:

$$E(T) = \int_0^{\infty} E(\nu, T) d\nu = \int_0^{\infty} E(\lambda, T) d\lambda, \quad (2)$$

Якщо інтервали  $d\nu$  і  $d\lambda$  стосуються однієї й тієї самої частини спектра, то

$$E(\nu, T) d\nu = E(\lambda, T) d\lambda, \quad (3)$$

Оскільки  $\nu\lambda = c$ , то  $d\nu = -\frac{c}{\lambda^2} d\lambda$  і вираз (3) набуває вигляду

$$E(\lambda, T) = \frac{c}{\lambda^2} E(\nu, T), \quad (4)$$

Знак мінус опустимо, оскільки він означає, що зі збільшенням частоти довжина хвилі зменшується, а нас цікавить абсолютне значення величини.

*Поглинальна здатність тіла*  $A(T)$  показує, яку частину падаючого випромінювання поглинає тіло при заданій температурі.

*Спектральна поглинальна здатність тіла*  $A(\nu, T)$  або  $A(\lambda, T)$  показує, яку частину падаючого випромінювання в інтервалі частот  $\nu, \nu + d\nu$  ( $\lambda, \lambda + d\lambda$ ) тіло поглинає при заданій температурі. Тіла, які здатні повністю поглинати падаюче на них випромінювання всіх довжин хвиль при будь-яких температурах, називають абсолютно чорними. Поняття абсолютно чорного тіла введено у 1859р. Г. Кірхгофом. Для таких тіл поглинальна здатність

$$A(T) = A(\nu, T) = A(\lambda, T) = 1.$$

За аналогією поглинальної здатності вводять поняття *відбивної здатності*  $R(T)$  і *спектральної відбивної здатності тіла*  $r(\nu, T)$  або  $r(\lambda, T)$ . Між спектральною об'ємною густиною енергії рівноважного випромінювання і спектральною випромінювальною здатністю абсолютно чорного тіла існує такий зв'язок:

$$u(\nu, T) = \frac{4}{c} E(\nu, T). \quad (5)$$

Абсолютно чорних тіл у природі немає. Найкращим наближенням до абсолютно чорного тіла є замкнута порожнина, у стінці якої зроблено малий отвір (рис. 1). Якщо стінки порожнини непрозорі, то за досить малих розмірів отвору в порожнині встановиться випромінювання, яке майже не відрізняється від рівноважного. Через отвір стінки порожнини виходитиме практично таке ж випромінювання, яке б випромінювалось при тій же температурі абсолютно чорною поверхнею, форма і розміри якої однакові з отвором порожнини. Основною властивістю абсолютно чорного тіла є незалежність характеру випромінювання і поглинання від природи та стану поверхні тіла.

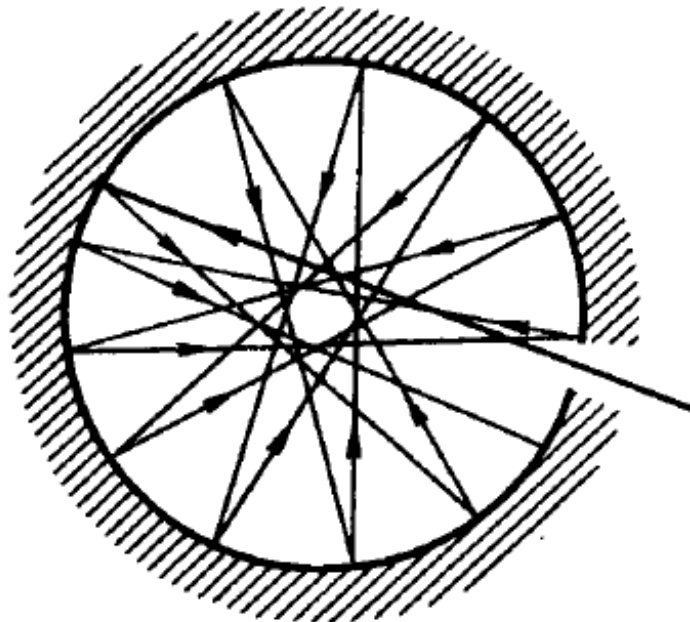


Рис. 1 Модель абсолютно чорного тіла

Випромінювальна здатність і розподіл енергії випромінювання за довжинами хвиль реальних тіл залежать від температури, хімічного складу і фізичною стану тіла та його поверхні. Незважаючи на прояв індивідуальних особливостей різних тіл, їх випромінювання має підлягати певним загальним законам. Одним із таких фундаментальних законів теорії тепловою випромінювання є закон, встановлений Г. Кірхгофом у 1859р. і названий його ім'ям. Закон Кірхгофа являє собою кількісне узагальнення встановленого емпірично П. Прево (1751-1839) у 1809 р. правила, яке формулюється так: якщо

поглинальні здатності тіл різні, то будуть різними і їх випромінювальні здатності.

Закон Кірхгофа ґрунтується на другому законі термодинаміки і встановлює залежність між випромінювальною і поглинальною здатностями тіл, а саме: відношення випромінювальної здатності тіла при деякій частоті і температурі до поглинальної здатності цього тіла при тій же частоті і температурі не залежить від природи тіла і являє собою універсальну функцію частоти і температури, яка дорівнює випромінювальній здатності абсолютно чорного тіла при тій же частоті і температурі, тобто

$$\frac{E(\nu, T)}{A(\nu, T)} = f(\nu, T), (6)$$

де  $f(\nu, T)$  - функція Кірхгофа.

Якщо позначити спектральну випромінювальну здатність абсолютно чорною тіла через  $e(\nu, T)$ , то через неї закон Кірхгофа запишеться так:

$$\frac{E(\nu, T)}{A(\nu, T)} = \frac{e(\nu, T)}{1} = f(\nu, T), (7)$$

звідки випливає, що функція Кірхгофа є функцією розподілу енергії в спектрі випромінювання абсолютно чорною тіла.

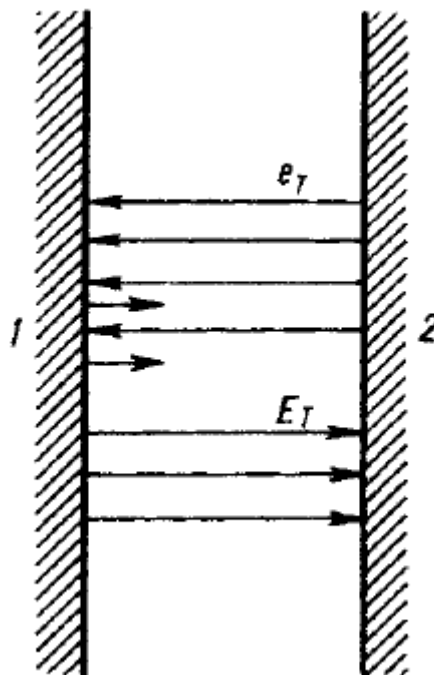


Рис. 2 Доведення закону Кірхгофа

Закон Кірхгофа для інтегральних значень випромінювальної і поглинальної здатностей тіла записується так:

$$\frac{E(T)}{A(T)} = \frac{e(T)}{1} = e(T). \quad (8)$$

Для доведення закону Кірхгофа уявімо довільне тіло у вигляді нескінченної однорідної пластини 1 (рис. 2). Паралельно до неї розміщено пластину 2, яка являє собою абсолютно чорне тіло. Позначимо випромінювальні і поглинальні здатності тіл 1 і 2 для заданої температури  $T$  відповідно  $E_T = E(T), A(T)$ ;  $e_T = e(T), a(T) = 1$ . Якщо між пластинами створити вакуум, то теплообмін між ними здійснюватиметься тільки за допомогою випромінювання. Через певний проміжок часу між тілами і випромінюванням, що заповнює простір між ними, настане термодинамічна рівновага і тіла матимуть однакову температуру  $T$ . При цьому одиниця поверхні кожного тіла за одиницю часу поглинатиме стільки енергії, скільки втрачає на випромінювання. Запишемо рівняння балансу для тіл 1 і 2 відповідно

$$E(T) = A(T)e(T); e(T) = E(T) + [1 - A(T)]e(T), \quad (9)$$

звідки одержуємо формулу (8).

Оскільки для реальних тіл  $A(T) < 1$ , то при одній і тій самій температурі абсолютно чорне тіло має найбільшу випромінювальну здатність. Із закону Кірхгофа також випливає, що будь-яке тіло при заданій температурі випромінює хвилі переважно тих довжин, які воно при тій самій температурі поглинає [17].

## 1.2 Закон Стефана - Больцмана. Закон Віна. Формула Релея - Джінса

Залежність випромінювальної здатності абсолютно чорною тіла від температури встановлює закон *Стефана – Больцмана*. Ця залежність була одержана експериментально у 1879 р. Й Стефаном (1835 - 1893), а в 1884 р. за допомогою методів термодинаміки теоретично виведена Л. Больцманом (1844-1906). В основу доведення покладена аналогія між термодинамічними властивостями рівноважного випромінювання й ідеального газу. Суть цієї аналогії полягає в тому, що як енергія ідеального газу, так і енергія рівноважного

випромінювання залежать тільки від температури; як газ, так і випромінювання здатні створювати тиск при взаємодії з поверхнею тіла. На основі цих уявлень з'ясовано, що випромінювальна здатність абсолютно чорного тіла пропорційна четвертому степеню температури

$$e(T) = \sigma T^4 . (10)$$

Формула (10) виражає закон Стефана – Больцмана:  $\sigma = 5,67 \cdot 10^{-8}$  Вт/( $\text{м}^2 \cdot \text{К}^4$ ) називається сталою Стефана - Больцмана. Її величина визначається за даними досліду.

Випромінювальна здатність реальних тіл може бути визначена на основі закону Кірхгофа

$$E(T) = A(T)\sigma T^4 . (11)$$

Закон Стефана - Больцмана встановлює залежність інтегральної випромінювальної здатності абсолютно чорного тіла від температури, але не розкриває характеру розподілу енергії випромінювання за довжинами хвиль або частотами, тобто конкретний вигляд функції Кірхгофа  $f(\nu, T)$  залишається не відомим. Як уже зазначалось, основним завданням теорії теплового випромінювання є встановлення конкретного вигляду цієї функції.

Криві розподілу енергії в спектрі випромінювання чорної сажі для різних температур одержав експериментально В. Ленглей у 1866 р. за допомогою болометра<sup>1</sup>. Його досліди показали, що для валових температур максимуми енергії випромінювання припадають на певні довжини хвиль, причому величина максимуму збільшується з підвищенням температури і зміщується у бік коротких довжин хвиль.

---

<sup>1</sup> Болометр (дав.-гр. βολή — промінь і μέτρον — міра) — прилад для вимірювання енергії випромінювання. Був винайдений Самуелем Пірпонттом Ленглі в 1878 р. Основний компонент болометра — дуже тонка пластинка (наприклад, з платини або іншого провідного матеріалу), зачорнена для кращого поглинання випромінювання. Через свою малу товщину пластинка під дією випромінювання швидко нагрівається і її опір підвищується. Для вимірювання малих відхилень опору пластинки її включають у вимірювальний міст, який балансують у відсутності засвічення.

Користуючись законами термодинаміки та електродинаміки, В. Він у 1893р. встановив характер залежності спектральної випромінювальної здатності абсолютно чорного тіла  $e(\nu, T)$  від частоти і температури. Було встановлено, що величина  $e(\nu, T)$  прямо пропорційна кубу частоти і є функцією відношення  $\nu/T$ , тобто

$$e(\nu, T) = \nu^3 F(\nu/T), \quad (12)$$

де  $F(\nu/T)$  - функція, конкретний вигляд якої В. Віну був невідомий. Незважаючи на те що одержана В. Віном залежність (12) не була встановлена в конкретному вигляді з неї виведено ряд важливих висновків. Так, за допомогою виразу (12) легко одержати підтвердження закону Стефана - Больцмана. У разі переходу від частоти  $\nu$  до  $\lambda$  вираз (12) набуває вигляду

$$e(\lambda, T) = \frac{c}{\lambda^2} e(\nu, T) = \frac{c^4}{\lambda^5} F\left(\frac{c}{\lambda T}\right), \quad (13)$$

звідки можна визначити довжину хвилі  $\lambda$ , при якій величина  $e(\lambda, T)$  досягає максимуму. При ньому було встановлено закон зміщення Віна

$$\lambda_{max} T = b, \quad (14)$$

де  $b = 2,89 \cdot 10^{-3} \text{ м} \cdot \text{К}$  - стала Віна.

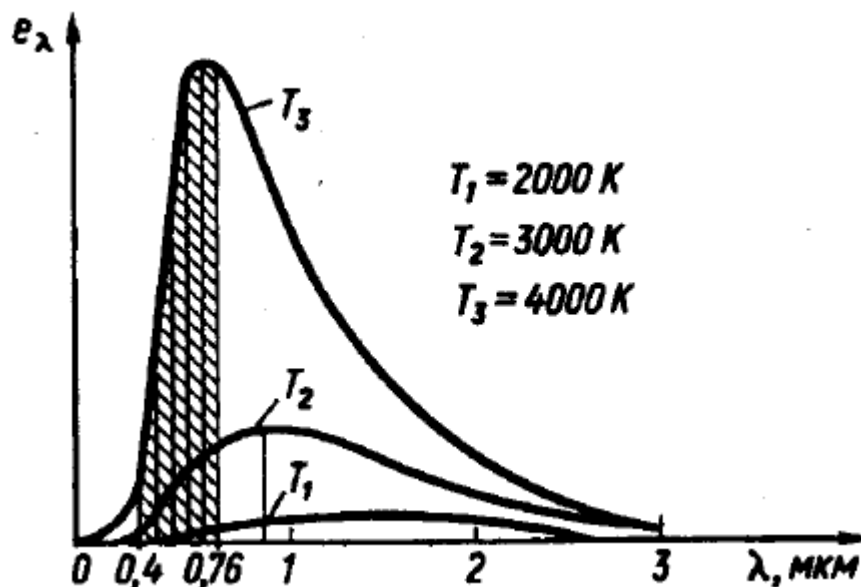


Рис. 3 Розподіл енергії у спектрі випромінювання чорного тіла для різних температур

За законом Віна (14) довжина хвилі  $\lambda_{max}$ , на яку припадає максимум спектральної випромінювальної здатності, при підвищенні температури зміщується в бік коротких хвиль.

Розподіл енергії у спектрі випромінювання чорного тіла для різних температур (рис. 3) показує, що для наведених температур максимуми кривих припадають на інфрачервону частину спектра. На рисунку також заштриховано площі під кривими розподілу для інтервалу довжин хвиль 0,4 до 0,76 мкм. Вони показують, яка саме частина енергії випромінювання припадає на видиму частину спектра при заданих температурах.

Користуючись формулами (13) і (14), знаходимо, що спектральна випромінювальна здатність для довжини хвилі, на яку припадає максимум випромінювання, пропорційна п'ятому степеню температури, тобто

$$e(\lambda_{max}, T) = \frac{c^4}{b^5} T^5 f\left(\frac{c}{b}\right) = aT^5, \quad (15)$$

де  $a = \frac{c^4}{b^5} T^5 f\left(\frac{c}{b}\right)$  - стала величина. Оскільки встановлення закономірностей теплового випромінювання відбувалося на основі законів термодинаміки, то конкретний вигляд функції розподілу енергії випромінювання абсолютно чорного тіла одержати неможливо.

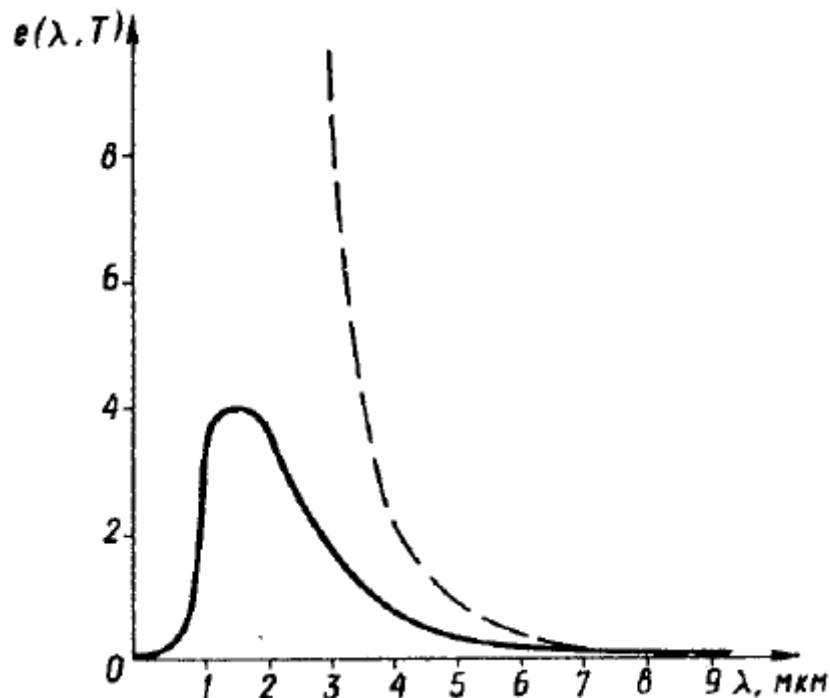


Рис. 4. Розбіжність експерименту з теорією у формулі Релея-Джінса

Щоб одержати закони теплового випромінювання, Дж. Релей і Дж. Джінс (1877-1946) застосували на відміну від своїх попередників, методи статистичної фізики, розглядаючи рівноважне випромінювання в порожнині як сукупність стоячих хвиль різних частот і довільних орієнтацій площин поляризації. Користуючись законом рівномірного розподілу енергії рівноважної системи за ступенями вільності, було встановлене співвідношення

$$e(\nu, T) = \left( \frac{2\pi\nu^2}{c^2} \right) kT, \quad (16)$$

яке називається формулою Релея - Джінса. Ця формула описує розподіл енергії в довгохвильовій частині спектра випромінювання, тобто для малих частот  $\nu$ . Для великих частот вона непридатна. Розбіжність між експериментальною кривою (суцільна лінія) і кривою, одержаною за допомогою формули Релея - Джінса (штрихова крива), видно з рис. 4. Крім того, спроба вивести закон Стефана - Больцмана за допомогою формули Релея - Джінса призвела до абсурдною результату, а саме:

$$e(T) = \int_0^{\infty} \frac{2\pi\nu^2}{c^2} kT d\nu = \infty.$$

Здоровий глузд заперечує такий результат. Ця ситуація, що створилась у теорії теплового випромінювання, образно була названа П. Еренфестом "ультрафіолетовою катастрофою" [17].

### 1.3 Формула Планка

Недоліки формул Віна і Релея-Джінса вказують на те, що для одержання функції розподілу енергії в спектрі випромінювання треба враховувати додаткові дані, що стосуються механізму випромінювання.

У 1900 р. М. Планк спочатку за результатами дослідних даних одержав емпіричний вираз функції  $e(\nu, T)$ , а потім теоретично і формулу на основі таких припущень: випромінювання є результатом коливання атомних лінійних вібраторів, які збуджують електромагнітні хвилі будь-яких частот подібно до

вібраторів Герца; вібратор випромінює енергію не безперервно, а певними порціями - квантами; енергія кванта, яка випромінюється вібратором, залежить від частоти випромінювання.

Формула Планка в змінних  $\nu$  і  $T$  має вигляд

$$e(\nu, T) = \frac{2\pi\nu^2}{c^2} \frac{h\nu}{e^{h\nu/kT} - 1}, \quad (17)$$

а в змінних  $\lambda$  і  $T$  на основі формули (4) відповідно

$$e(\lambda, T) = \frac{2\pi hc^2}{\lambda^5} \frac{1}{e^{hc/k\lambda T} - 1}, \quad (18)$$

де  $h = 6,626 \cdot 10^{-34}$  Дж · с - стала Планка.

Оскільки формула Планка справедлива для будь-яких частот і температур, то з неї мають впливати всі відомі закони випромінювання абсолютно чорною тіла (закон Стефана - Больцмана, закон зміщення Віна та ін.). Виведемо ці закони на основі формули Планка. Сумарна випромінювальна здатність абсолютно чорного тіла

$$e(T) = \int_0^\infty e(\nu, T) d\nu = \frac{2\pi h}{c^2} \int_0^\infty \frac{\nu^3 d\nu}{e^{h\nu/kT} - 1}. \quad (19)$$

Ввівши безрозмірну змінну інтегрування  $x = \frac{h\nu}{kT}$ , надамо виразу (19) такого вигляду:

$$e(T) = \frac{2\pi k^4}{c^2 h^3} T^4 \int_0^\infty \frac{x^3 dx}{e^x - 1} = \sigma T^4,$$

де  $\sigma = \frac{2\pi k^4}{c^2 h^3} \int_0^\infty \frac{x^3 dx}{e^x - 1}$ . Оскільки  $\int_0^\infty \frac{x^3 dx}{e^x - 1} = \frac{\pi^4}{15}$ , то

$$\sigma = \frac{2\pi^5 k^4}{15c^2 h^3} = 5,67 \cdot 10^{-8} \frac{\text{Вт}}{\text{м}^2 \text{К}^4}.$$

Отже, формула Планка дає змогу не тільки встановити закон Стефана-Больцмана, але й знайти величину сталої  $\sigma$ .

Знайдемо довжину хвилі, на яку припадає максимум спектральної випромінювальної здатності при заданій температурі. для цього треба розв'язати рівняння

$$\frac{de(\lambda, T)}{d\lambda} = 0. \quad (20)$$

З формули (18) маємо

$$\frac{\frac{hc}{kT\lambda^2} e^{hc/k\lambda T}}{e^{hc/k\lambda T} - 1} - \frac{5}{\lambda} = 0. \quad (21)$$

Ця умова виконується, коли

$$\frac{hc}{kT\lambda} e^{\frac{hc}{kT\lambda}} - 5e^{\frac{hc}{kT\lambda}} + 5 = 0. \quad (22)$$

Ввівши змінну  $x = \frac{hc}{kT\lambda}$ , одержуємо трансцендентне рівняння

$$xe^x - 5e^x + 5 = 0, \quad (23)$$

розв'язком якого є  $x = 4,965$ . Тоді

$$T\lambda_{max} = \frac{hc}{4,965k} = b = 2,898 \cdot 10^{-3} \text{ м} \cdot \text{К}. \quad (24)$$

Отже, з формули Планка одержуємо закон зміщення Віна і величину сталої Віна  $b$ .

Випромінювання всіх твердих тіл, рідин і газів характеризується нерівномірним розподілом енергії у спектрі випромінювання, а їхні інтегральні і спектральні поглинальні здатності завжди менші, ніж в абсолютно чорного тіла.

Якщо спектр випромінювання тіла суцільний, а розподіл енергії за довжинами хвиль подібний до розподілу енергії у спектрі абсолютно чорного тіла, то такі тіла називаються сірими. Для сірих тіл при заданій температурі

$$\frac{E(\lambda, T)}{e(\lambda, T)} = const = A(T). \quad (25)$$

Тіла, для яких умова (25) не виконується, називаються селективними. Випромінювання реальних тіл у деякій мірі має селективний характер. Найбільша селективність випромінювання характерна для газів, оскільки вони випромінюють хвилі у певних порівняно вузьких ділянках спектра.

Сірих тіл, так само, як і абсолютно чорних, у природі не існує. В інтервалі довжин хвиль від фіолетової до далекої інфрачервоної частини спектра для жодного тіла умова (25) не виконується. Однак для різних тіл у деяких частинах спектра умова (25) справджується з певною точністю [17].

Наявність безпосереднього зв'язку інтенсивності теплового випромінювання тіла з його температурою, обґрунтованої в законах теплового

випромінювання Стефана-Больцмана, Планка, Віна, дозволило здійснювати дистанційне вимірювання (контроль) температури шляхом реєстрації теплового випромінювання. Ця особливість техніки отримала назву термографія.

## 2. Способи отримання термографічних зображень

### 2.1 Цифровий контактний термограф

В Донецькому фізико-технічному інституті ім. А.А. Галкіна НАН України, була представлена нова розробка цифрового контактного термомамографа (рис. 5), забезпеченого сканером з 36 термочутливими інтелектуальними датчиками, що дозволяють вимірювати температуру з роздільною здатністю  $0,06\text{ }^{\circ}\text{C}$  і перетворювати її в цифровий код з подальшою комп'ютерною обробкою температурних карт поверхонь молочних залоз, отриманих послідовним позиціонуванням сканера. Принцип дії заснований на контактному вимірюванні температури поверхні шкіри одночасно великою кількістю датчиків. Датчики організовані у вигляді квадратної матриці (сканера) розміром  $6 \times 6\text{ см}$ . Одномоментний вимір температури 36 напівпровідниковими цифровими датчиками проводиться з ділянки молочної залози площею  $\sim 40\text{ см}^2$ . Послідовним переміщенням сканера по поверхні молочної залози складається мозаїчна картина розподілу температури - температурна карта поверхні шкіри або термомамограма. Отримана термомамограма з'являється на екрані комп'ютера в кольоровому або чорно-білому варіантах.



Рис. 5 Цифровий контактний термомамограф

Термомомаграф дозволяє визначити: температуру кожного датчика в отриманій термограмі, тобто температуру ділянки шкіри площею 1 см<sup>2</sup>; різниця температури між симетричними точками лівої і правої молочної залози; середню температуру кожної позиції, тобто середню температуру по всіх датчиків сканера при одноразовому його позиціюванні на якій-небудь ділянці залози; середню температуру залози; розмір і розташування областей максимальних температурних аномалій.

У аналітичну комп'ютерну програму включено до 30 ознак для кількісної оцінки отриманих даних при термомомаграфії. Термомомаграф дозволяє сформувати інтерпольовані термограми, вивести на принтер протокол, який містить відомості про пацієнтку, термограму, результати статистичної обробки термограми, а також висновок лікаря. Прилад має роздільну здатність 0,06 °С і дозволяє відображати температуру в діапазоні від 20 ° до 38 °С.



Рис. 6 Блок-схема цифрового контактного термомомаграфа

Блок-схема приладу представлена на рис. 6. Датчики являють собою мікропроцесори, здатні вимірювати температуру з дискретністю 1/16 °С, протягом секунди перетворювати температуру в цифровий код і пересилати цей код в комп'ютер для обробки і зберігання. Теплова інерційність датчиків 10-15 с, вони не вимагають калібрування. На термограмі можливо відобразити температуру до 1152 точок. Ці властивості датчиків дозволяють проводити обробку отриманої інформації в реальному масштабі часу, формалізувати діагностичні параметри і автоматизувати первинну діагностику[18-19].

Одним з недоліків такою методу є нерівномірність контакту термочутливої пластинки з епітелієм, знижує точність методу.

## 2.2 Фоточутливі приймачі, які застосовуються в термографах

### 2.2.1 Болومتر

Болومتر - сенсор, що змінює опір в залежності від температури.

Вловлюючи ці зміни, електронна схема формує вихідний сигнал, який представляє собою діаграму розподілу температур по матриці. Для цього на матрицю періодично подається струм зміщення і вимірюється величина отриманої напруги. Найбільш поширеним матеріалом для виготовлення болOMETричних детекторів в даний час є полікристалічний окис ванадію. Завдяки прогресу технологій виробництва напівпровідникових компонентів, виробники тепловізорів на мікроболومتر зуміли ефективно вирішити проблему розкиду параметрів окремих пікселів при мініатюризації приладів (це основне джерело власних шумів матриці). Сучасні неохолоджувані камери здатні вловити різницю температур в 0,07 К. Одна з технічних тонкощів в проектуванні подібних матриць відеокамери внутрішнього купольного типу - розрахунок розміру ніжок осередки: вони повинні бути досить малі, щоб забезпечити задані теплоізоляційні властивості, і досить великі для того, щоб встигати знизити температуру води протягом інтервалу сканування пікселів (в іншому випадку на наступне вимірювання опору буде впливати результат попереднього).

Крім відсутності вимог до охолодження, мікроболومتر мають ще одну суттєву перевагу перед фотоелектричними сенсорами. Діаграма їх спектральної чутливості має практично плоску форму у всьому діапазоні частот ІЧ-променів. Сучасні мікроболOMETричні матриці мають часом реакції в декілька сотих часток секунди, що дозволяє їм забезпечувати формування стандартного телевізійного сигналу в реальному масштабі часу зі стандартною частотою кадрів. Це частково компенсує їх відставання від охолоджуваних детекторів по чутливості. Незважаючи на відсутність необхідності в охолодженні, багато різновидів

мікроболометрів потребують температурної стабілізації, яка реалізується за допомогою термовентилятора. Як правило, сенсор і вентилятор укладені в герметичний корпус, термічно ізольований від решти компонентів камери. Це дозволяє гарантувати, що на сенсор потраплятимуть лише ІЧ-промені, що випускаються спостережуваними об'єктами. Ряд виробників користуються схемами електронної температурної компенсації, що дозволяють виключити застосування термовентиляторів. Порівняно недавно виник і абсолютно новий вид детекторів інфрачервоного випромінювання на базі мікро-кантильоверної технології. За чутливості вони в десять разів перевершують мікроболометр, наближаючись до теоретичних меж чутливості матеріалів до інфрачервоних променів.

### 2.2.2 Приймачі ІЧ-зображення на основі термомеханічних нанорозмірних мембран

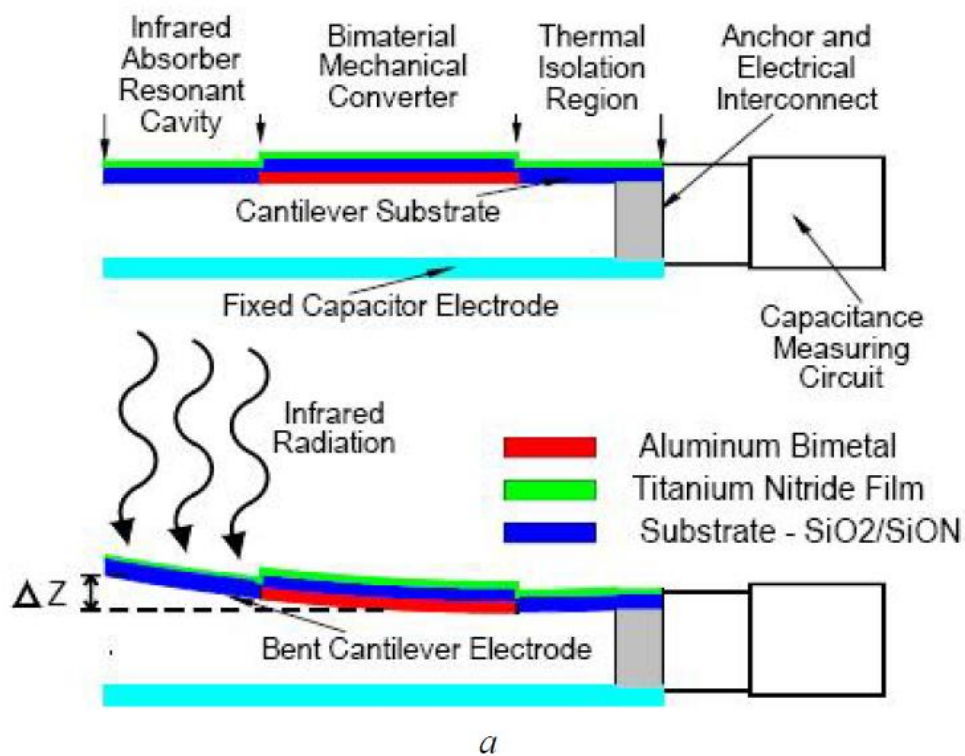
Останнім часом поряд з традиційними приймачами болометричного типу активно розробляються і досліджуються неохолоджувані ІЧ-фотоприймачі на основі МОЕМС і НОЕМС - мікро- і нанооптоелектромеханічних елементів і систем.

Великий інтерес представляють, зокрема, детектори на біматеріальних шарах. Оцінки граничної чутливості таких приймачів [20, 21], показали, що вона може досягати кількох мілікельвінів NETD (Noise equivalent temperature difference (різниця температур, еквівалентна шуму)), що можна порівняти з граничними характеристиками фотоприймачів з глибоким охолодженням.

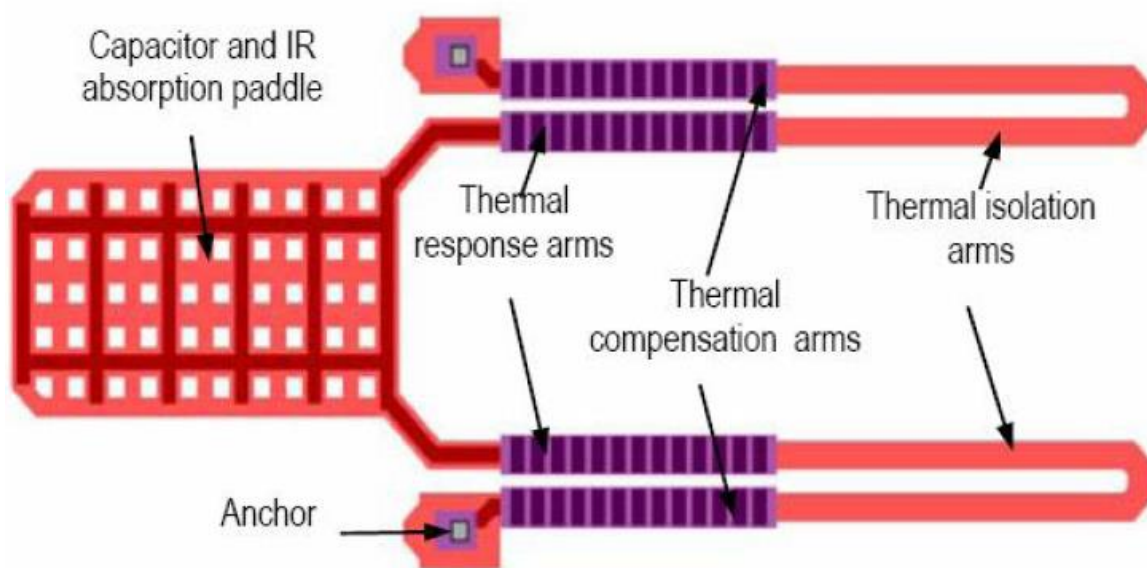
Пристрої з параметрами, близькими до граничних, можуть бути реалізовані в даний час на базі високорозвиненої технології МОЕМС / НОЕМС [22-26].

Принцип функціонування таких приймачів ІЧ-зображення базується на реєстрації вигину тонких двошарових біморфних (або багатошарових - мультиморфної) мембран під впливом нагріву ІЧ-випромінюванням на базі відмінності коефіцієнтів термічного розширення матеріалів шарів. У

представлених в літературі розробках мембрани виконуються зазвичай у вигляді тонких (менше ніж 1 мкм) балок - мікрокантільверів. На рис. 7, а, б схематично представлені принцип дії та чутливий елемент біморфного МОЕМС-детектора [24].



*a*



*б*

Рис. 7 Схема:

а) - функціонування детектора; б) – чутливого елемента

Відхилення кантильовера може реєструватися ємнісним методом. На рис. 8 представлена одна з можливих схем ємнісної реєстрації (мостова схема) [24].

Можливо також використання схем оптичного зчитування [23, 25].

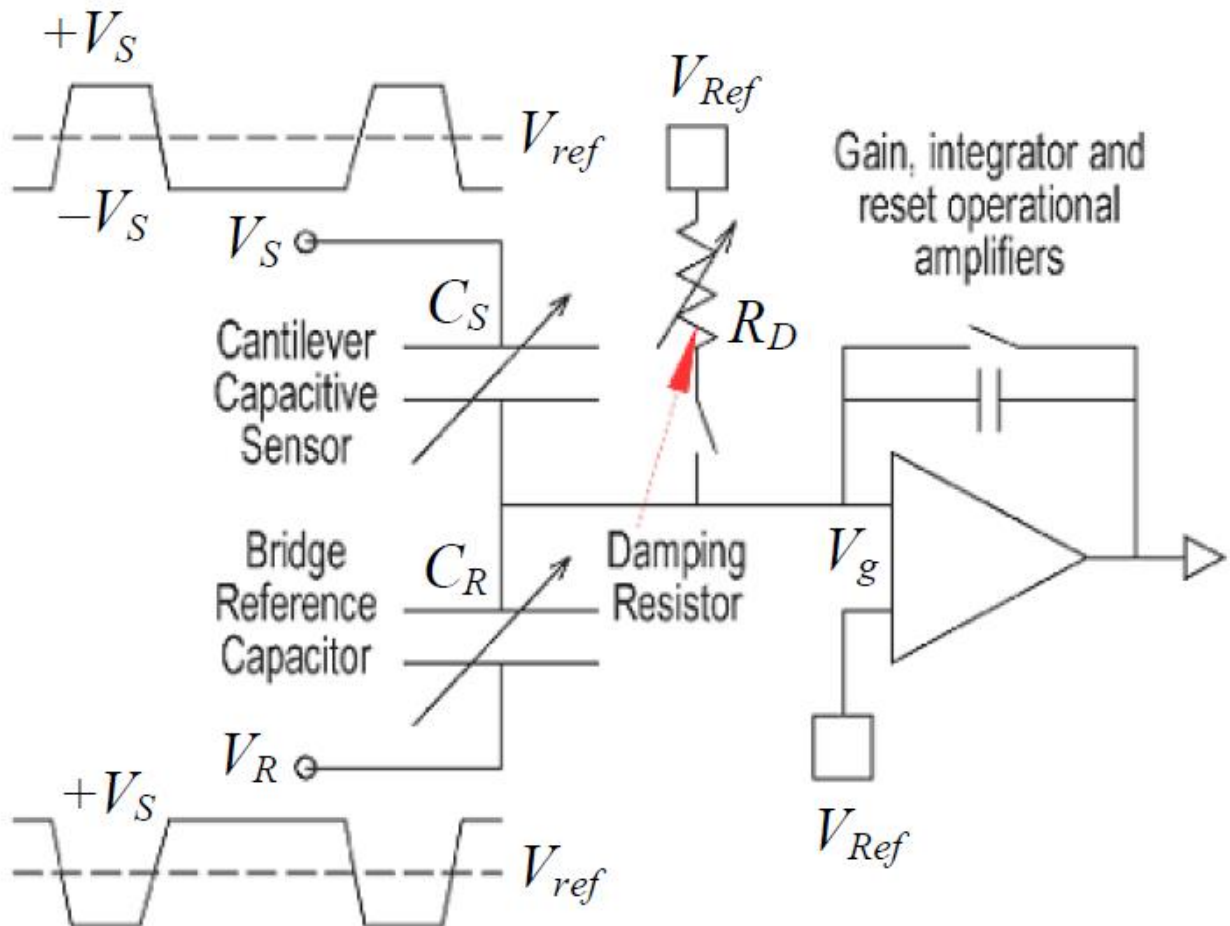


Рис. 8. Мостова схема реєстрації відхилення кантильовера

МОЕСМ-детектор розглянутого типу служить в якості елемента матричного ПЧ-приймача зображення. В роботі [24] повідомляється про створення і випробуванні приймача такого типу з числом елементів  $160 \times 120$  і розміром пікселя  $50 \text{ мкм}$  з ємнісною системою зчитування на чіпі в вакуумованій мікробірці.

Продемонстровані аналогічні пристрої з оптичною системою зчитування [23] - це гібридні пристрої. Як формувач відеосигналу в них використовуються кремнієві матриці ПЗС; система зчитування може включати фільтр просторових частот або будується по диференціальній схемі з поляризацією випромінювання.

### 2.2.3 Фотоприймачі на основі теллуридів кадмію і ртуті

Переважає більшість сучасної оптико-електронної апаратури інфрачервоного діапазону спектра створюється з використанням фотоприймачів на основі твердого розчину теллуридів кадмію і ртуті (КРТ) (рис. 9), незважаючи на технологічну складність і високу ціну цього матеріалу, Основними причинами є унікальні властивості  $Cd_xHg_{1-x}Te$  [27]:

- КРТ з оптимальними властивостями для реєстрації ІЧ випромінювання в основних вікнах прозорості земної атмосфери може бути отриманий простою зміною співвідношення кадмію і ртуті в цьому твердому розчині: діапазон 1-2.5мкм при  $x \approx 0.4$ , 3-5мкм при  $x \approx 0.3$ , 8-12мкм при  $x \approx 0.2$ ;
- Фотоприймачі з КРТ на всі спектральні діапазони можуть виготовлятися за єдиною планарною технологією;
- КРТ може бути як n-типу провідності з малою концентрацією носіїв, так і р-типу з високою структурною досконалістю Це забезпечує можливість створення як високоефективних фоторезисторів на діапазон до 15 і більше мкм, так і фотодіодів з високою швидкістю;
- КРТ може вирощуватися як у вигляді монокристалів, так і у вигляді тонких епітаксіальних шарів на прозорих в необхідній області спектра підкладках, що забезпечує можливість створення великоформатних діодних фоточутливих матриць;
- Фотоприймачі на основі КРТ мають набагато більшу стійкість до проникаючого випромінювання завдяки більшій дефектності структури, ніж прилади на основі антимоніду індію, кремнію та інших напівпровідникових матеріалів.

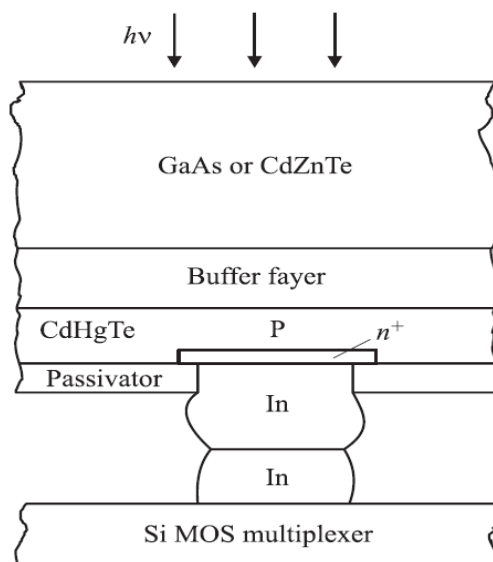


Рис. 9 Будова фоточутливого елемента.

### 3. Способи перетворення інфрачервоного випромінювання в зображення на екрані монітора

#### 3.1 Будова тепловізора

Принцип роботи всіх тепловізорів описується наступним алгоритмом (рис. 10):

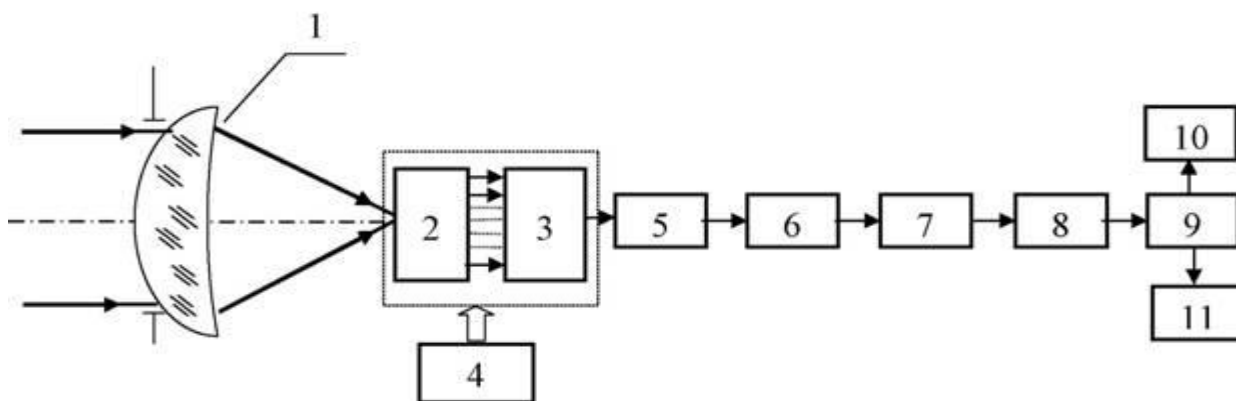


Рис. 10 Схема тепловізора:

1 - оптична система; 2 - фокальна матриця з підсилювачами; 3 - мультиплексор; 4 - система охолодження; 5 - коректор неоднорідності характеристик чутливих елементів; 6 - аналого-цифровий перетворювач; 7 - цифровий коректор неоднорідності; 8 - коректор непрацюючих осередків; 9 - формувач зображення; 10 - дисплей; 11 - цифровий вихід.

- Об'єктив тепловізора формує на сенсорі температурну карту (або карту різниці потужності випромінювання) всієї спостережуваної в полі зору області.
- Процесор і інші електронні компоненти конструкції зчитують дані з матриці, обробляють їх і формують на дисплеї приладу зображення, що є візуальною інтерпретацією цих даних, яке безпосередньо або через окуляр розглядає спостерігач.

### 3.2 Опис діючої програми в термографі з охолоджуваною матрицею сенсорів

Цією програмою оснащений термограф розроблений спільно з Інститутом фізики напівпровідників ім. В.Є. Лошкарьова.

Процедура для ініціалізації основної форми програми, та задання початкових параметрів для компонент розміщених на основній формі:

```

procedure TForm1.Init(Sender: TObject);
var t: integer;
begin
  PosX.Ctl3D:=False;
  PosY.Ctl3D:=False;
  PosT.Ctl3D:=False;
  PosX.Text:=' ';
  PosY.Text:=' ';
  PosZ.Text:=' ';
  PaletteGood:=false;
  T30:= 225;
  T35:= 370;
  dt_k:= (T35-T30)/(35-30);
  MyPal.InitPalettes;

```

```
MyPal.MakeMainPalette(0);
MyPal.MakeUsedPalette(0,4095);
b:=TBitmap.Create;
b.pixelformat:=pf24bit;
b.width:=Main_Image.Width;
b.height:=Main_Image.Height;
b2:=TBitmap.Create;
b2.pixelformat:=pf24bit;
b2.width:=Main_Image.Width;
b2.height:=Main_Image.Height;
spectr1:=TBitmap.Create;
spectr1.pixelformat:=pf24bit;
spectr1.width:=PaletPanel.Width;
spectr1.height:=PaletPanel.Height;
spectr2:=TBitmap.Create;
spectr2.pixelformat:=pf24bit;
spectr2.width := Spectr2Panel.Width;
spectr2.height:= Spectr2Panel.Height;
DrawSpectr;
MoveMouse:=false;
MovePalet:= false;
ComboBox1.ItemIndex:=0;
ComboBox2.ItemIndex:=1;
hair.y0:=0;
hair.x0:=0;
hair.hor:=RadioButton1.Checked;
hair.vert:=RadioButton2.Checked;
t:=MyFrame.Init;
end;
```

Процедура для звільнення ресурсів пам'яті та виходу з програми:

```
procedure TForm1.FormDestroy(Sender: TObject);
begin
    b.free;
    b2.free;
    spectr1.free;
    spectr2.free;
    MyFrame.Close;
end;
```

Процедура для зміни та регулювання кольорової гами на термограмі:

```
Procedure TForm1.DrawSpectr;
var x, y : integer;
    k : longint;
    p : PARGB;
begin
    for y:=1 to 10 do
        begin
            p:= spectr1.scanline[y];
            for x:=0 to 255 do
                begin
                    k:=x*16;
                    p[x].r:= MyPal.UsedPalette.r[k];
                    p[x].g:= MyPal.UsedPalette.g[k];
                    p[x].b:= MyPal.UsedPalette.b[k];
                end;
            end;
        PaletPanel.canvas.draw(0,0,spectr1);

        for y:=0 to Spectr2Panel.Height-1 do
```

```

begin
  p:= spectr2.scanline[y];
  k:= Round((Spectr2Panel.Height-1 - y)*$1000/Spectr2Panel.Height);
  if MyPal.UsedPalette.Pal_16 then
    k:=k and $F00 + $80;

    for x:=1 to 9 do
      begin
        p[x].r:= MyPal.MainPalette.r[k];
        p[x].g:= MyPal.MainPalette.g[k];
        p[x].b:= MyPal.MainPalette.b[k];
      end;
    end;
  Spectr2Panel.canvas.draw(0,0,spectr2);
  Series6.Clear;
  With Chart3.RightAxis do
    begin
      Automatic := False;
      AutomaticMaximum := False;
      AutomaticMinimum := False;
      Minimum := (30+(MyPal.UsedPalette.min-T30)/dt_k);
      Maximum := (35+(MyPal.UsedPalette.max-T35)/dt_k);
    end;
  Image6.Left:=(MyPal.UsedPalette.max div 16)+26;
  Image5.Left:=(MyPal.UsedPalette.min div 16)+23;
end;

```

Процедура для відображення термограми на формі, та створення і відображення зрізу термограми на осцилографі:

```

procedure TForm1.PutFrame(Sender: TObject);

```

```
var x, y, z, z2 : integer;
    p, p_sub : PARGB;
    for_shl : byte;
begin
    for_shl:=SpinEdit2.Value;
begin
    if hair.hor then
        begin
            Series4.Clear;
            for x:=0 to 255 do
                begin
                    z:=dat.d[hair.y0,x];
                    Series4.Add(z,"clTeeColor);
                end;
            Series5.Clear;
            z:=dat.d[hair.y0,hair.x0];
            Series5.AddXY(hair.x0,z,"clTeeColor);
        end;

    if hair.vert then
        begin
            Series4.Clear;
            for y:=0 to 289 do
                begin
                    z:=dat.d[y,hair.x0];
                    Series4.Add(z,"clTeeColor);
                end;
            Series5.Clear;
            z:=dat.d[hair.y0,hair.x0];
            Series5.AddXY(hair.y0,z,"clTeeColor);
```

```

    end;
end; { ScopeCheck.Checked}

for y:=0 to 289 do
    begin
        p:= b.scanline[y];
        p_sub := b2.scanline[y];

        if (y=hair.y0) and hair.hor then
            begin
                for x:=0 to 255 do
                    begin
                        z:=dat.d[y,x];
                        p[x].r:= MyPal.UsedPalette.r[z] xor $FF;
                        p[x].g:= MyPal.UsedPalette.g[z] xor $FF;
                        p[x].b:= MyPal.UsedPalette.b[z];
                    end;
                    continue;
                end;
                for x:=0 to 255 do
                    begin
                        z:=dat.d[y,x]+8;
                        if CheckBox4.Checked then
                            begin
                                z2:= dat.d[y,x] - dat2.d[y,x]+20;
                                if z2<0 then z2:=0;
                                z2:= z2 shl for_shl;
                                p_sub[x].r:= MyPal.UsedPalette.r[z2];
                                p_sub[x].g:= MyPal.UsedPalette.g[z2];
                                p_sub[x].b:= MyPal.UsedPalette.b[z2];
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;

```

```

    end;
    p[x].r:= MyPal.UsedPalette.r[z];
    p[x].g:= MyPal.UsedPalette.g[z];
    p[x].b:= MyPal.UsedPalette.b[z];

    if hair.vert and (x=hair.x0) then
    begin
        p[x].r:= MyPal.UsedPalette.r[z] xor $FF;
        p[x].g:= MyPal.UsedPalette.g[z] xor $FF;
        p[x].b:= MyPal.UsedPalette.b[z];
    end;
end;
end;
Main_Image.canvas.draw(0,0,b);

```

```

if CheckBox4.Checked then
begin
    Image3.canvas.draw(0,0,b2);
    Image2.Picture:= Image3.Picture;
end;
Button3Click(Sender);
end;

```

Процедура для відслідковування координати курсора на термограмі і вивід координат в текстові компоненти під термограмою, та побудова зрізу термограми на основі цих координат:

```

procedure TForm1.Main_ImageMouseMove(Sender: TObject; Shift:
TShiftState; X,
    Y: Integer);
var x1, k : integer;

```

```
begin
  if not MoveMouse then exit;
  if y>289 then y:=289;
  if y<0 then y:=0;
  if x>255 then x:=255;
  if x<0 then x:=0;

  PosX.Text:=' '+IntToStr(X);
  PosY.Text:=' '+IntToStr(Y);

  hair.y0:=y;
  hair.x0:=x;

  Series4.Clear;

  if hair.hor then
    for x1:=1 to 254 do
      begin
        k:=dat.d[y,x1];
        Series4.Add(k,",clTeeColor);
      end;

  if hair.vert then
    for x1:=1 to 289 do
      begin
        k:=dat.d[x1,x];
        Series4.Add(k,",clTeeColor);
      end;

  Series5.Clear;
```

```

k:=dat.d[y,x];
if hair.hor then
  Series5.AddXY(x,k,"clTeeColor)
else Series5.AddXY(y,k,"clTeeColor);
PosZ.Text:=' '+IntToStr(k);

PosT.Text:=' '+FloatToStrF(30+(k-T30)/dt_k,ffFixed,4,2);
end;

```

Процедура для зберігання термограми на жорсткий диск :

```

procedure TForm1.SaveImageClick(Sender: TObject);
var F : file;
    b : boolean;
begin
  b:=Timer1.Enabled;
  Timer1.Enabled:=false;
  if SaveDialog1.Execute then
  begin
    AssignFile(F, SaveDialog1.FileName);
    Rewrite(F, 8);
    with dat.header do
    begin
      image_type[0]:='T';
      image_type[1]:='M';
      image_type[2]:='G';
      image_type[3]:='3';
      format[2]:=PZS_1;
      format[3]:=PZS_h;
    end;
    BlockWrite(F, Dat, 18624);
  end;

```

```

    CloseFile(F);
end;
Timer1.Enabled:=b;
end;

```

Процедура для зберігання термограми у вигляді BMP файлу:

```

procedure TForm1.SaveBitmapClick(Sender: TObject);
var SourceRect, DestRect: TRect;
    x, y, dx, dy : integer;
    b: boolean;
    Bitmap: TBitmap;
begin
    b:=CheckBox1.Checked;
    CheckBox1.Checked:=false;
    PutFrame(Sender);
    Application.ProcessMessages;
    x:=444;
    y:=295;
    dx:=256;
    dy:=290;
    Bitmap := TBitmap.Create;
    Bitmap.Width:=444;
    Bitmap.Height:=295;

    SourceRect := Rect(GroupBox1.Left+1,      GroupBox1.Top+14,
GroupBox1.Left+4+dx, GroupBox1.Top+16+dy );
    DestRect := Rect(1,1,x-62,y);
    Bitmap.Canvas.CopyRect(DestRect,Form1.Canvas,SourceRect);
    x:=Chart3.Left+GroupBox1.Left;
    y:=Chart3.Top+GroupBox1.Top+8;

```

```

dx:= Chart3.Width;
dy:= Chart3.Height-16;
SourceRect:=Rect(x,y, x+dx, y+dy);
DestRect := Rect( 380, 1, 380 + dx, dy-2);
Bitmap.Canvas.CopyRect(DestRect,Form1.Canvas, SourceRect);

if SaveDialog1.Execute then
begin
    Bitmap.SaveToFile(ChangeFileExt(SaveDialog1.FileName, '.BMP'));
    ShowMessage(ChangeFileExt(SaveDialog1.FileName, '.BMP') + ' is saved');
end;
Bitmap.Free;
CheckBox1.Checked:=b;
end;

```

Процедура для зчитування попередньо збереженої термограми на жорсткому диску:

```

procedure TForm1.Load1Click(Sender: TObject);
var F : file;
    i, k, m, ymin, ymax : integer;
    xmax , xmin, x,y ,counts : integer;

begin
    MoveMouse:=false;
    if OpenFileDialog1.Execute then
    begin
        Timer1.Enabled:=false;

        AssignFile(F, OpenFileDialog1.FileName);
        Reset(F, 8);

```

```

BlockRead(F, MyDat, 21669, counts);
CloseFile(F);

for i:=0 to 4095 do plot8[i]:=0;
ymin:=0; ymax:=0;
xmin:=0; xmax:=0;

if counts=21669 then
  for i:=0 to 289 do
    for k:=0 to 255 do
      dat.d[i,k]:=MyDat.Dark[i,k]
  else if counts=18624 then
    for i:=0 to 289 do
      for k:=0 to 255 do
        dat.d[i,k]:=MyDat.Frame[i,k] ;

for i:=0 to 289 do
  for k:=0 to 255 do
    begin
      inc(plot8[dat.d[i,k]]);
    end;

minp:=0;
maxp:=4095;
for x:=1 to 4095 do
  if plot8[x]=0 then minp:=x else break;
for x:=4095 downto minp+1 do
  if plot8[x]=0 then maxp:=x else break;
if maxp<4084 then maxp:=maxp+10;
if minp>10 then minp:=minp-10;

```

```

AutoCheckClick(Sender);

Series1.Clear;
for i:=0 to 4095 do
begin
Series1.Add(plot8[i]/742.4,",clTeeColor);
end;

Timer2.Enabled:=true;
SpeedButton6.Enabled:=true;
StatusBar1.SimpleText:=' '+OpenDialog1.FileName+' ';
end;
end;

```

### 3.3 Удосконалення програмного забезпечення термографа ТВС-300

В тепловізорах медичного призначення нерідко використовують матриці, які мають пошкоджені пікселі, це пов'язане з тим, що бездефектні матриці використовують, в основному, для приладів військового призначення. При роботі тепловізора в режимі “осцилограф” температурна шкала не завжди відповідає фактичним значенням реальної температури. Інтерес представляє отримання зображень, в яких візуалізуються ділянки з мінімальною або максимальною температурою. Важливими факторами в розширенні діагностичних можливостей інфрачервоної термографії є визначення площі патологічної зони та створення тривимірного зображення об'єкту.

Завдання полягає в розробці та впровадженні вдосконаленого програмного забезпечення досліджень біологічних об'єктів, що допоможе вирішити визначені проблеми.

В роботі використовувався вітчизняний охолоджуваний матричний термограф з базовою тестовою програмою для тепловізійної системи “Thermo Visio”. Його технічні характеристики наведено в таблиці.

Таблиця

Точність вимірювання температури	0,07°C
Кут поля зору	40°
Число елементів матриці	240 (H) x 320(V)
Розмір елемента	50 мкм (H) x 33 мкм (V)
Частота кадру	25 Гц
Діапазон спектральної чутливості	2 ÷ 5,3 мкм
Вихідний сигнал камери	цифровий
Напруга живлення	220 В ± 10% з частотою 50 Гц ± 1%
Система охолодження	заливний азотний кріостат

### 3.3.1 Розробка програми по видаленню фонових шумів

Розглянемо термографічне зображення руки (червоними колами виділено ділянку з пошкодженими пікселями), яка візуалізується як зона з підвищеною температурою (див. рис. 11).

Для того, щоб видалити такі ділянки, знаходимо середнє значення температури на цій термограмі та середньоквадратичне відхилення. Наступним кроком визначаємо та залишаємо тільки ті значення температури, які не перевищують три середньоквадратичних відхилення (правило 3-х сигм ( $3\sigma$ )). Практично всі значення нормально розподіленої випадкової величини лежать в інтервалі  $[\bar{x} - 3\sigma; \bar{x} + 3\sigma]$  [28]. Це означає, що з достовірністю 99,7%, значення нормально розподіленої випадкової величини лежить у вказаному інтервалі [29]. Розрахунки виконувались з використанням стандартних формул:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (26) \quad \sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (27)$$

Термограми об'єктів до та після обробки представлені на рис. 11.

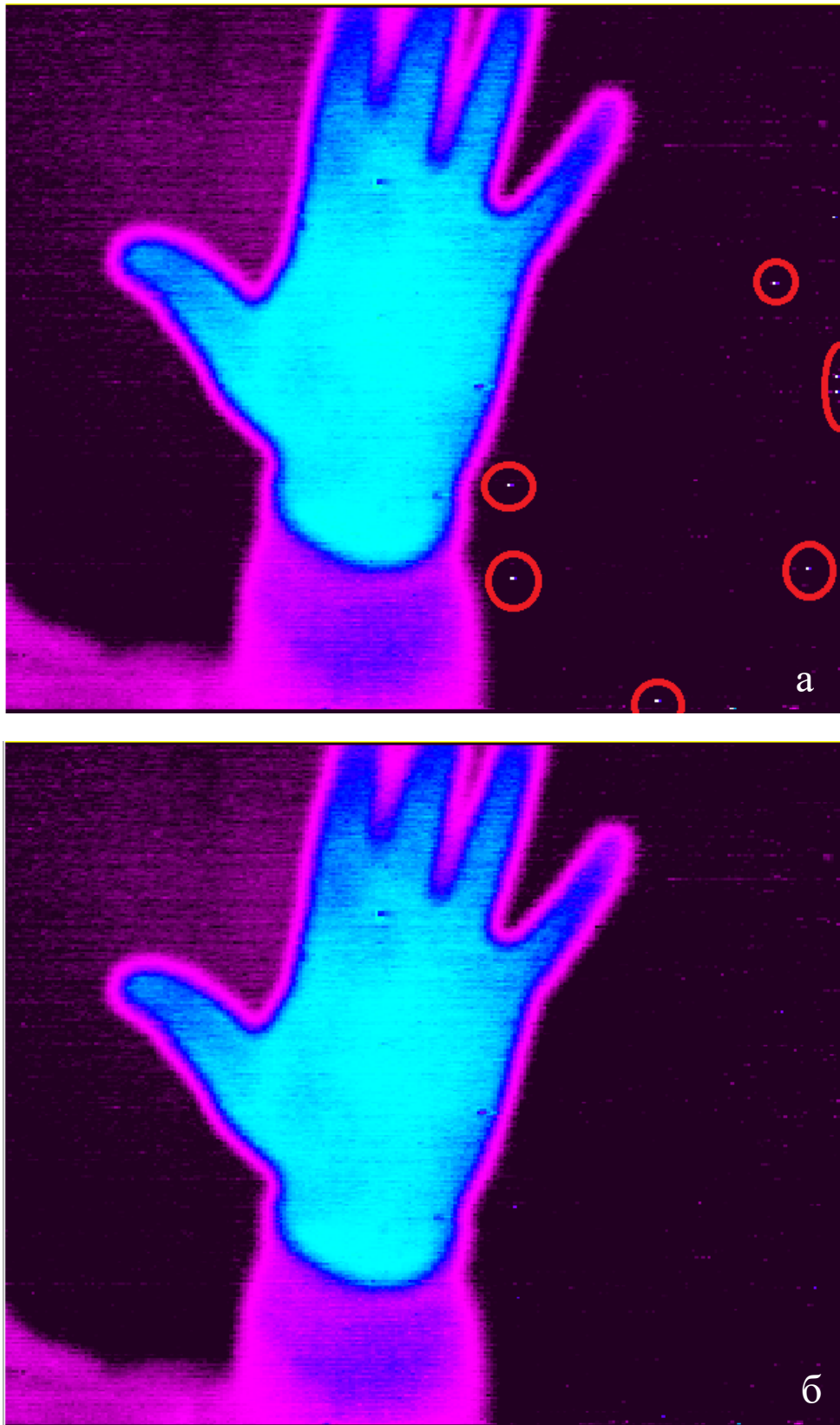


Рис. 11. Термограми об'єктів до (а) та після (б) обробки.

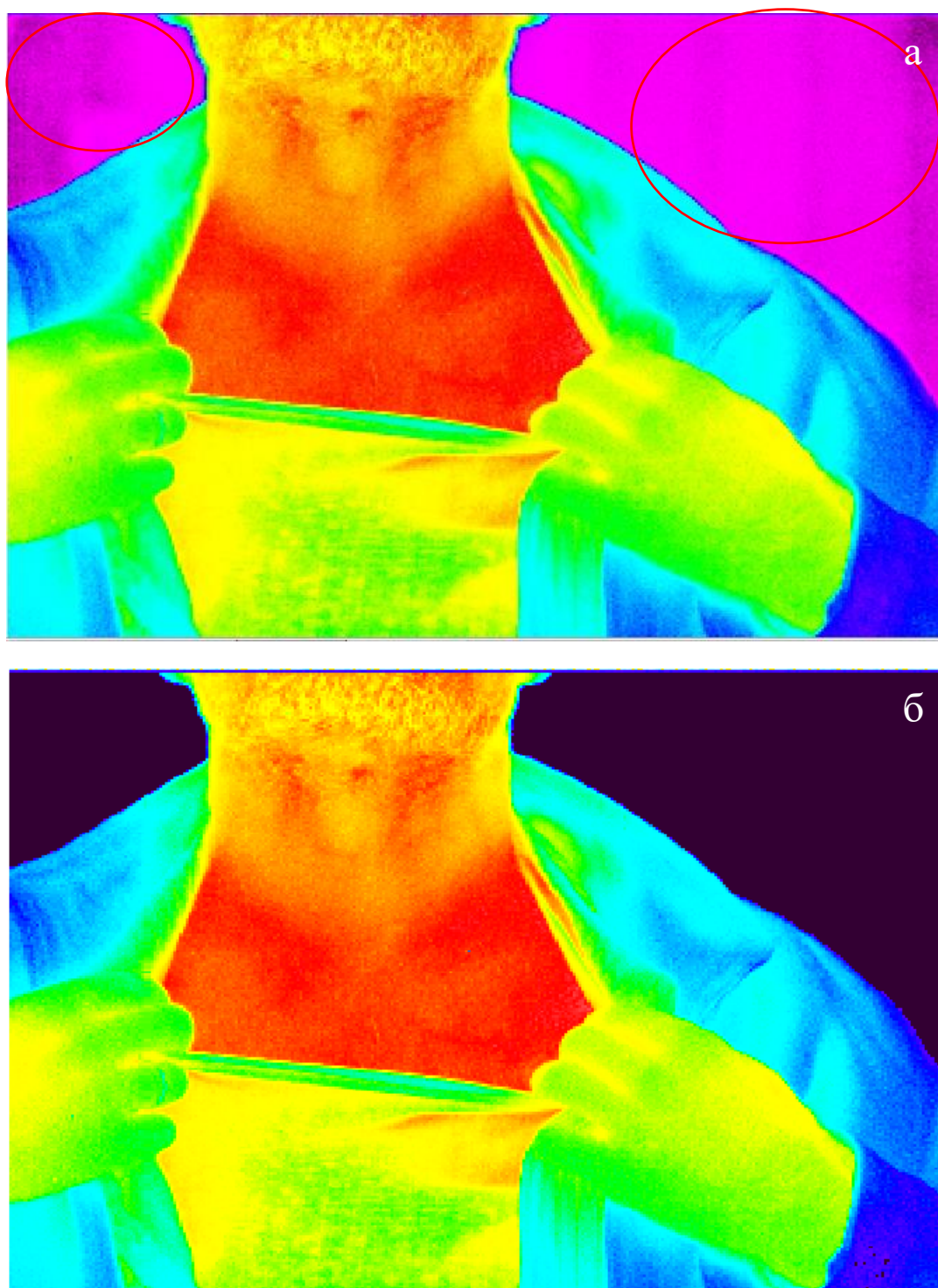


Рис.12. Термограми до (а) та після (б) усунення пошкоджених точок та фонових шумів.

При аналізі термографічних зображень досить чітко спостерігаємо наявність фонових шумів. Для того, щоб їх прибрати і, таким чином, збільшити контрастність зображення, знаходимо ділянки з мінімальною температурою. В

області мінімальної температури завжди буде міститися фоновий шум, так як зони з мінімальною температурою прилеглі до кордону нижньої границі діапазону вимірювання температури термографом, який впливає на якість термографічного зображення. Температура по периметру термографічного зображення має більшу температуру фонових шумів на термограмі. Вилучаючи ділянки з мінімальною температурою, ми видаляємо фоновий шум.

Результат застосувань двох алгоритмів - прибирання пошкоджених точок та фонових шумів (на рис. виділено колом) наведено на рис. 12.

### 3.3.2 Розробка програми по виділенню окремих температурних шарів

Під час термографічного дослідження виникає потреба виділити на термограмі ділянки з мінімальною, максимальною або проміжною температурою. Розроблений програмний код для виділення на термографічному зображенні таких ділянок.

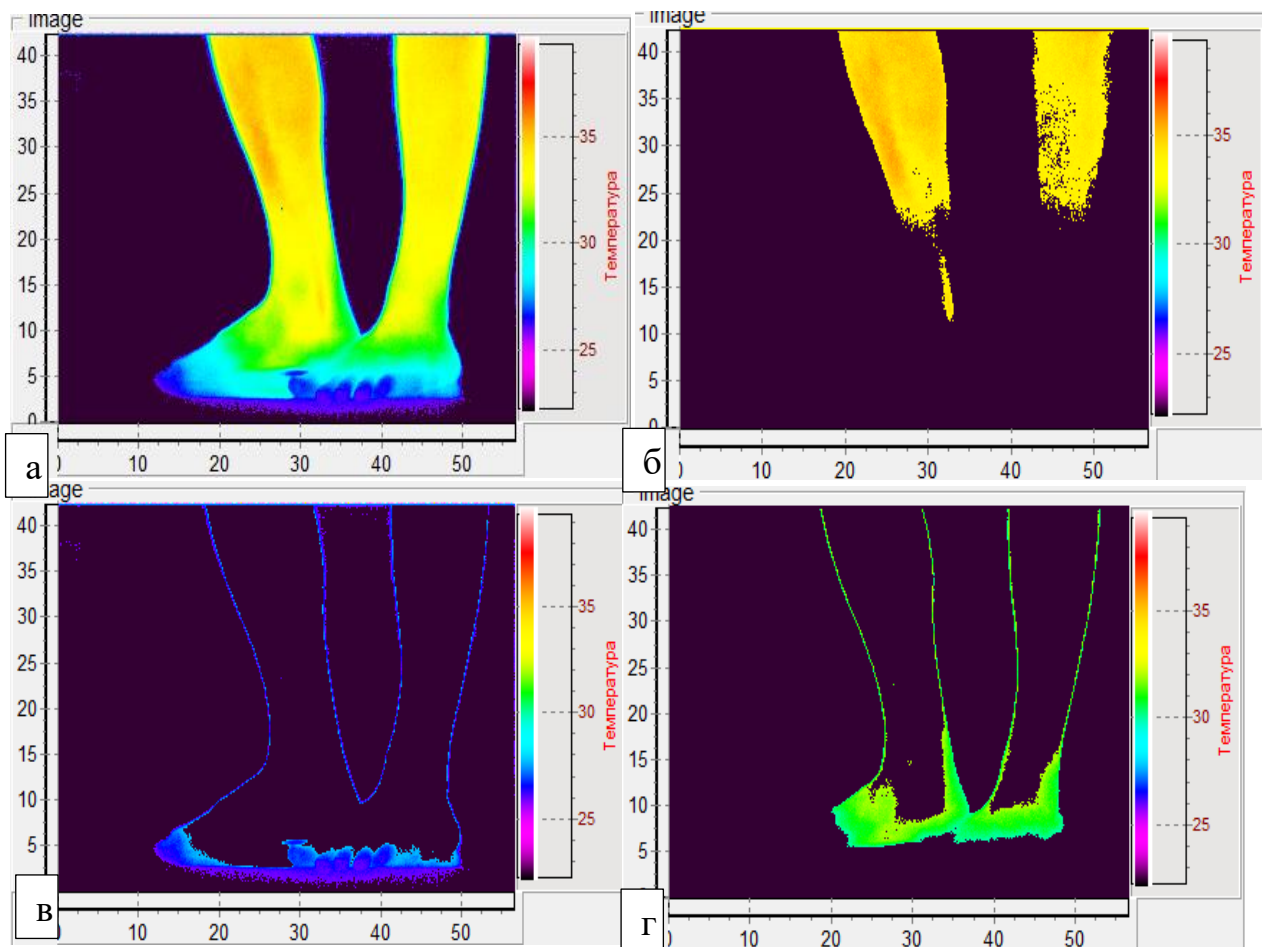


Рис.3. Термограми до (а) та після виділення максимальної (б) мінімальної (в) та проміжної (г) температури.

Термографічне зображення досліджуваного об'єкту до та після застосування програмного коду по виділенню ділянок з максимальною, мінімальною та проміжною температурою показано на рис. 13.

### 3.3.3 Розробка програми по переведенню температури в реальні значення

Для спрощення аналізу термограм при роботі термографа в режимі “осцилограф” відносні значення температур відобразимо в реальні їх значення. Вирішення цього питання здійснювалось наступним чином: візьмемо два відомих значення температури на термограмі, застосуємо рівняння прямої, проведеної через дві точки  $\frac{t-t_1}{t_2-t_1} = \frac{y-y_1}{y_2-y_1}$  (28), де  $t$  – реальні значення температури,  $y$  - відносні значення температури.

Виконані відповідні програмні зміни дозволили при роботі термографа в режимі “осцилограф” вимірювати реальні значення температури в зонах інтересу.

Результат осцилографічних зображень до (а) та після (б) програмного використання коду для переведення відносних значень температури в реальні значення в градусах Цельсія приведені на рис. 14.

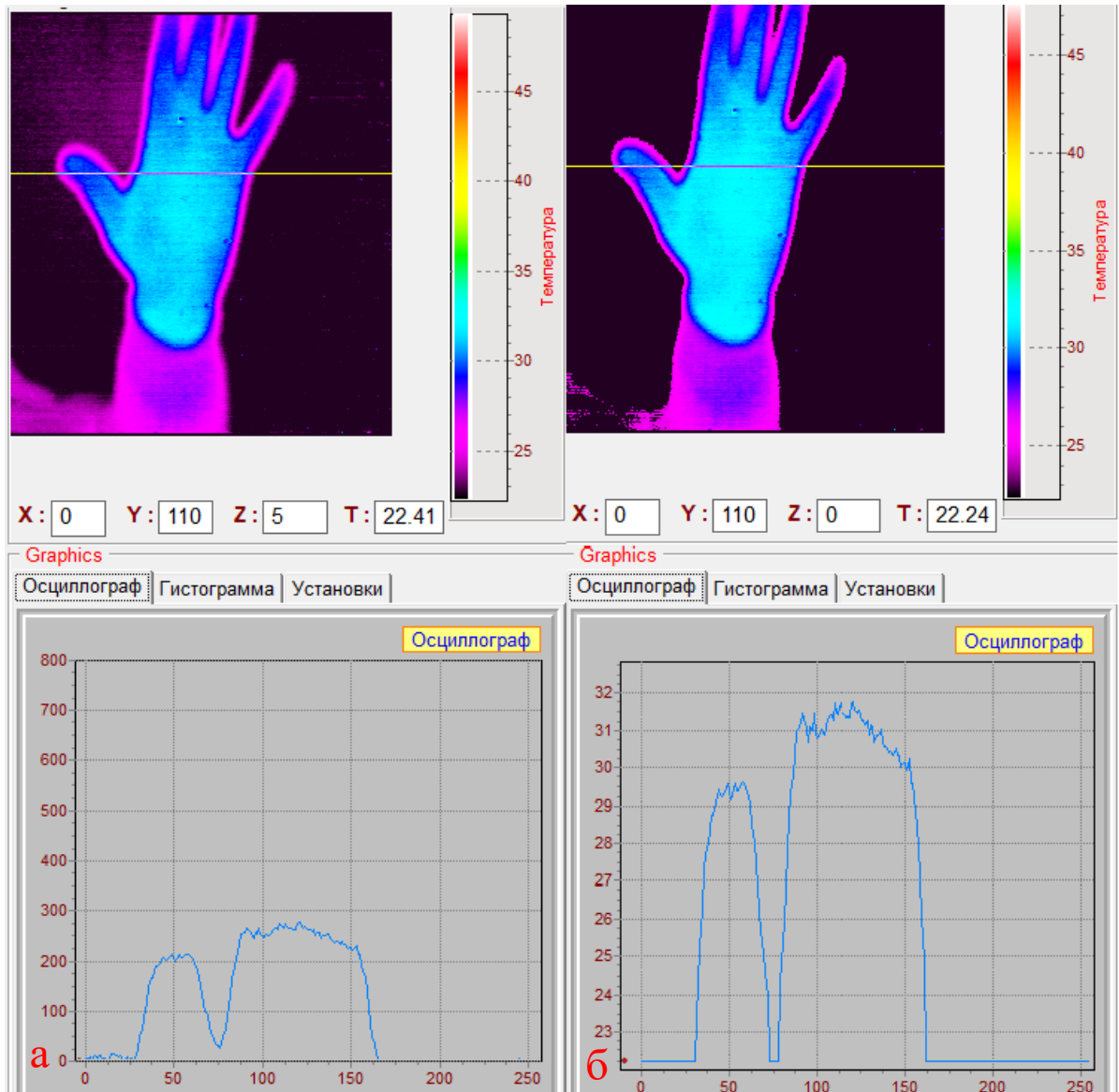


Рис.14. Термограми зображень та осцилограми температурного профілю в відносних та реальних значеннях температури.

### 3.3.4 Розробка програми для визначення площі ділянки об'єкту досліджень

При виконанні термографічних досліджень біологічного об'єкту виникає потреба у визначенні площі ураженої ділянки, що має особливе значення під час операційних втручань.

Розроблено також спосіб визначення площі зони інтересу та розроблено відповідне програмне забезпечення.

Об'єктив розглядаємо як тонку лінзу [30] й при розрахунках використовуємо формулу тонкої лінзи. Формула для визначення збільшення лінзи:

$$H = \frac{h \cdot F \cdot f}{f(f-F)} = \frac{F \cdot h}{f-F} \quad (29)$$

де:  $F$  - фокусна відстань,  $f$  - відстань між зображенням і лінзою,  $h$  - зображення предмета,  $H$  - дійсна величина предмета.

Для знаходження площі ураженої ділянки, виконаємо розрахунок збільшення лінійних розмірів зображення, яке ми спостерігаємо на екрані монітора. Матриця має розміри 240x320 пікселів (px). Збільшення по осі OY визначаємо за формулою:  $H_1 = \frac{F \cdot 240}{f-F}$  (30); а по осі OX -  $H_2 = \frac{F \cdot 320}{f-F}$  (31).

При цьому формула для визначення збільшення об'єкту для буде мати вигляд:  $S_p = \frac{H_1 \cdot H_2}{76800}$  (32).

Порахувавши кількість пікселів, які задіяні у зображенні ( $N$ ), знайдемо площу об'єкта ( $S$ ):  $S = S_p \cdot N$ . (Питання підрахунку кількості пікселів незадіяних від задіяних в отриманому зображенні, вирішено програмним шляхом). Використовуючи формулу для визначення збільшення розмірів зображень на матриці, побудуємо шкалу для отримання фактичних розмірів досліджуємої ділянки (рис. 15), що візуалізується на екрані монітора.

В якості прикладу розглянемо термограму пацієнта з порушенням функції дихання (рис. 15). Вимірювання площі перерізу повітряного потоку, що вдихається пацієнтом, має важливе діагностичне значення для лікаря-оториноларинголога у визначенні ступені тяжкості патологічного процесу.

Виділяємо за допомогою рамки зону ураження та вираховуємо за вищенаведеними формулами площу шляхом натискання на кнопку «Площадь объекта в рамке» (див. рис. 16).

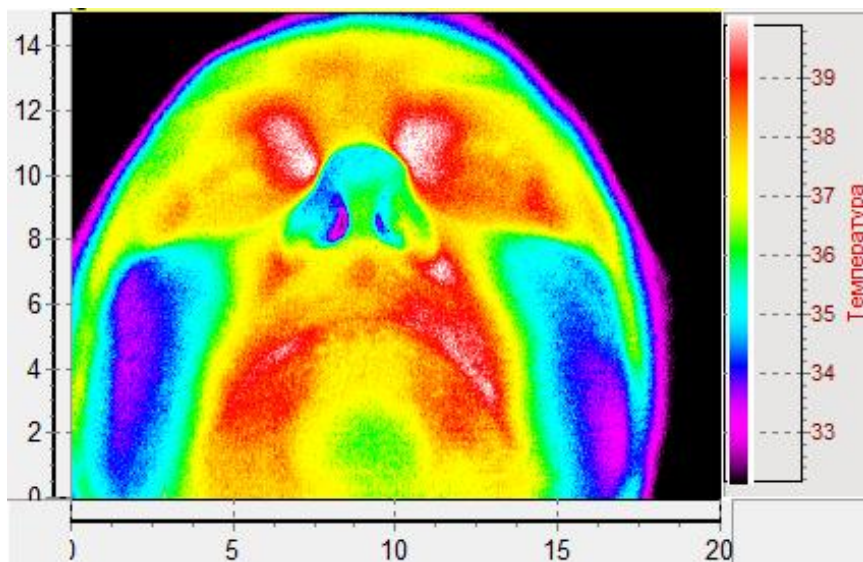


Рис. 15. Термограма пацієнта з порушенням функції дихання.

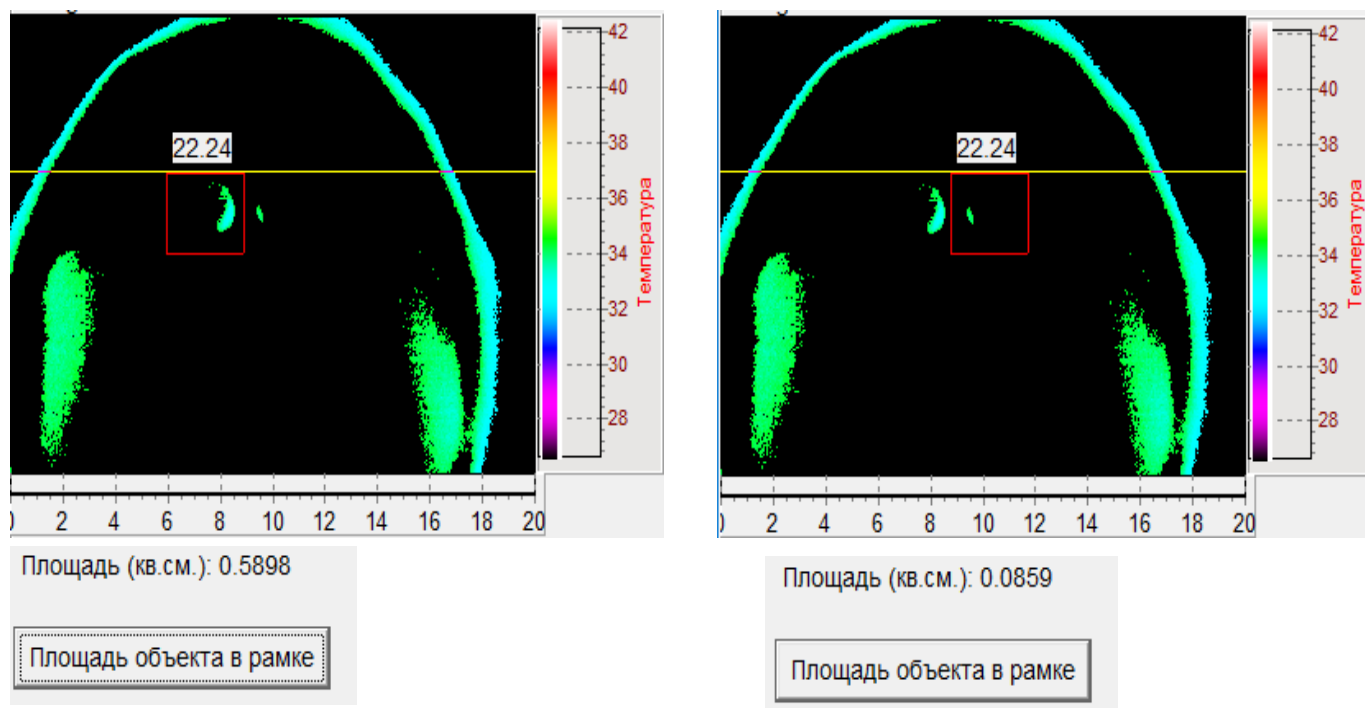


Рис.16. Визначення площі зони інтересу.

### 3.3.5 Розробка програми для побудови 3D графіку

Отримання тривимірного зображення дозволить підвищити інформативність та уявлення про патологічні відхилення всього досліджуваного об'єкту. В режимі «осцилограф» отримуємо термозріз площини перерізу зони інтересу (див. рис. 17,а). За допомогою 3D графіку отримуємо сумарний термозріз по всій поверхні досліджуваного об'єкту, що розширює можливості для аналізу. Це дозволить візуально порівнювати значення температури не тільки за кольором на термограмі, а й за висотою ліній на графіку по осі OZ (див. рис. 17,б).

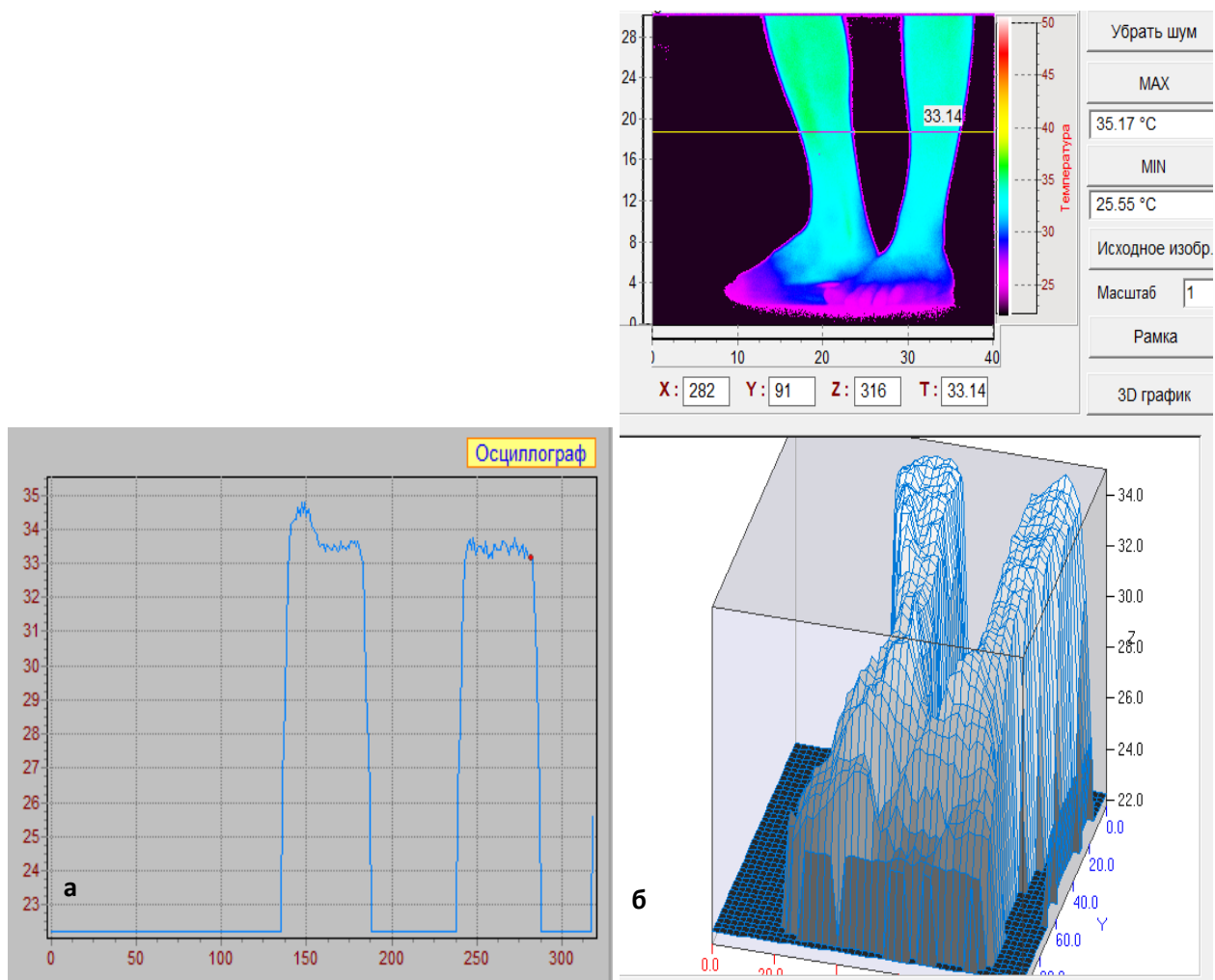


Рис. 17. Термозріз на термограмі в режимі «осцилограф» (а) та сумарний термозріз по всій поверхні досліджуваного об'єкту (б).

#### 4. Розробка оновленого програмного забезпечення

##### 4.1 Опис нової програми

Програмний код процедури по видаленню пошкоджених пікселів:

```
procedure visualNoise;
```

```
var min,max,i,k,i1,k1,range,n, avpoint, denom:integer ;
```

```
average, dispersion,sum, stDeviation :real;
```

```
begin
```

```
min:=dat.d[0,0];
```

```
max:=dat.d[0,0];
```

```
sum:= 0;
```

```
n:= 0;
```

```
// вкладені цикли для визначення min та max температури та розрахунок суми для
```

```
// формул (26, 27);
```

```
for i:=0 to 239 do
```

```
for k:=0 to 319 do
```

```
begin
```

```
if min > dat.d[i,k] then min:= dat.d[i,k];
```

```
if max < dat.d[i,k] then max:= dat.d[i,k];
```

```
sum:=sum + dat.d[i,k];
```

```
inc(n);
```

```
end;
```

```
range:=max-min;
```

```
average:= sum/n;
```

```
sum:=0;
```

```
for i:=0 to 239 do
```

```
for k:=0 to 319 do
```

```
begin
```

```
sum:= sum + (dat.d[i,k]*dat.d[i,k] - average*average);
```

```
end;
```

```

dispersion:= sum / n;
stDeviation:= sqrt(dispersion);
for i:=0 to 239 do
  for k:=0 to 319 do
    begin
      if (dat.d[i,k]<(average - 3*stDeviation)) or (dat.d[i,k] > (average +
3*stDeviation)) then
        begin
          {===== average value around non-standart point =====}
          avpoint:=0;
          denom:=0;
          for i1:=i-1 to i+1 do
            for k1:=k-1 to k+1 do
              begin
                if (dat.d[i1,k1]>(average - 3*stDeviation)) and (dat.d[i1,k1] <
(average + 3*stDeviation))
                  and not((i1=i) and (k1=k)) then
                    begin
                      avpoint := avpoint + dat.d[i1,k1];
                      inc(denom);
                    end;
                end;
              if denom>0 then avpoint:=round(avpoint/denom);
              {=====}
              dat.d[i,k]:= avpoint;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

Програмний код процедури по прибиранню фонових шумів:

```

procedure TForm1.SpeedButton10Click(Sender: TObject);
var min, max, i,k,range:integer ;
percent :real;
begin
min:=10000;
max:=dat.d[0,0];
  for i:=0 to 239 do
    for k:=0 to 319 do
      begin
        if (min > dat.d[i,k]) and (dat.d[i,k] <> 0) then min:= dat.d[i,k];
        if max < dat.d[i,k] then max:= dat.d[i,k];
      end;
    range:=max-min;
    percent:=0.11*range;
// визначення фонової температури (значення нижчі за це значення вважається
//шумом);
    min := min + Round(percent);
    for i:=0 to 239 do
      for k:=0 to 319 do
        begin
          if dat.d[i,k] < min then dat.d[i,k] := 0;
        end;
      Area();
    end;
end;

```

Програмний код процедур по виділенню температурних шарів:

```

procedure MinMax(extreme : String);
var min,max,i,k,range:integer ;
percent:real;
initialValue: array [0..239,0..319] of integer;

```

```

begin
min:=10000{dat.d[0,0]};
max:=dat.d[0,0];
  for i:=0 to 239 do
    for k:=0 to 319 do
      begin
        if (min > dat.d[i,k]) and (dat.d[i,k] <> 0) then min:= dat.d[i,k];
        if max < dat.d[i,k] then max:= dat.d[i,k];
        initialValue[i,k]:= dat.d[i,k];
      end;
    end;
  range:=max-min;
  percent:=0.2*range;

  if extreme = 'max' then
    for i:=0 to 239 do
      for k:=0 to 319 do
        begin
          if (dat.d[i,k] < (max-percent)) then dat.d[i,k]:=0;
        end;
      end;
    end;
  if extreme = 'min' then
    for i:=0 to 239 do
      for k:=0 to 319 do
        begin
          if (dat.d[i,k] > (min+percent)) then dat.d[i,k]:=0;
        end;
      end;
    end;
  end;

  Процедура для кнопки по виділенню максимальної температури:
  procedure TForm1.SpeedButton8Click(Sender: TObject);
  begin
    MinMax('max');
  end;

```

```
Area();
```

```
end;
```

Процедура для кнопки по виділенню мінімальної температури:

```
procedure TForm1.SpeedButton9Click(Sender: TObject);
```

```
begin
```

```
  MinMax('min');
```

```
  Area();
```

```
end;
```

Процедура для кнопки по виділенню проміжної температури:

```
procedure TForm1.Main_ImageDblClick(Sender: TObject);
```

```
  var i,j, temp:integer;
```

```
begin
```

```
  temp := strtoint(PosZ.Text);
```

```
  for i:=0 to 239 do
```

```
    for j:=0 to 319 do
```

```
      if (temp-30>dat.d[i,j]) or (dat.d[i,j]> temp+30) then dat.d[i,j] := 0;
```

```
    Area();
```

```
end;
```

Програмний код для переведення температурної шкали в реальні значення:

```
if ((30+(max-T30)/dt_k)*1.01) <= ((30+(min-T30)/dt_k)*0.99) then
```

```
  begin
```

```
    Chart2.LeftAxis.Maximum:= (30+(max-T30)/dt_k);
```

```
    Chart2.LeftAxis.Minimum:= (30+(min-T30)/dt_k);
```

```
  end
```

```
else
```

```
  begin
```

```
    Chart2.LeftAxis.Maximum:= (30+(max-T30)/dt_k)*1.01;
```

```
    Chart2.LeftAxis.Minimum:= (30+(min-T30)/dt_k)*0.99;
```

```
  end;
```

Код для лейбла над курсором з реальними значеннями температури:

```
{ Label with temperature above the cursor }
form1.label_temp.caption:=FloatToStrF((30+(dat.d[y,x]-
T30)/dt_k),ffFixed,4,2); // label on image
    form1.label_temp.left:=x+4;
    form1.label_temp.top:=y-4;
```

Програмний код процедури для визначення площі ділянки:

```
procedure Area();
var area, i, j :integer;
    Focus, distToObj, sizeFrameH,sizeFrameV,SquarePixel :real;
begin
area := 0;
for i:=0 to 239 do
for j:=0 to 319 do
    if dat.d[i,j]> 0 then inc(area);

Focus := StrToFloat(Form1.Edit1.Text);
distToObj := StrToFloat(Form1.Edit8.Text);
if Focus <> 0 then
begin
    sizeFrameH := 320*(distToObj - Focus)/Focus;
    sizeFrameV := 240*(distToObj - Focus)/Focus;
    SquarePixel := sizeFrameV*sizeFrameH/76800;
end
else
begin
    sizeFrameH := 0;
    sizeFrameV := 0;
end;
```

```

With Form1.Chart4.BottomAxis do
begin
  Automatic := False;
  AutomaticMaximum := False;
  AutomaticMinimum := False;
  Minimum := 0;
  if sizeFrameH > Minimum then Maximum := sizeFrameH
  else Maximum := Minimum;
  Increment := trunc(Maximum*0.1);
end;
With Form1.Chart5.LeftAxis do
begin
  Automatic := False;
  AutomaticMaximum := False;
  AutomaticMinimum := False;
  Minimum := 0;
  if sizeFrameV > Minimum then Maximum := sizeFrameV
  else Maximum := Minimum;
  Increment := Form1.Chart4.BottomAxis.Increment;
end;
Form1.label11.Caption:= 'Площадь (px): ' + floattostr(area);
if Focus <> 0 then
  Form1.label_area.Caption:= 'Площадь (кв.см.): ' +
    floattostr(SimpleRoundTo(SquarePixel * area,-4))
  else Form1.label_area.Caption:= 'Площадь (кв.см.): infinity';
end;

```

Процедура для побудови 3D графіку:

```

procedure plot3d(startv, starth: real; x,y:integer);

```

```

var i,k:integer ;
new_temp,max,min :real;
begin;

form1.P3d.SuppressPaint := true;
form1.P3d.GridMat.Resize(47,42); // max size of matrix : 2000 elements

max:=0;
min:=1000;

for i:=1 to 47 do
begin;
for k:=1 to 42 do
begin;
new_temp:= (30+(dat.d[trunc(k*startv+y),trunc(i*starth+x)]-T30)/dt_k);
if (new_temp > max) then max:=new_temp;
if (new_temp < min) and (new_temp<>0) then min:=new_temp;
form1.P3d.GridMat[i,k] := new_temp;
end;
end;
// визначення min та max температури проводиться для коректного підбору
//діапазону температур по осі OZ;
max:=Trunc(max)+1;
form1.P3D.ColorScaleHigh:=max;
if min < 28 then
form1.P3D.SetRange(0,100,0,100,min-0.5,max)
else if (min >= 28) then
form1.P3D.SetRange(0,100,0,100,min-4,max);
form1.P3d.SuppressPaint := false;
end;

```

## 4.2 Інструкції по використанню нових функцій програми

1. Відкриваємо файл програми, після відкриття програми (рис. 18) натискаємо на зображення папки (1) в “меню” програми та вибираємо термограму для аналізу в діалоговому вікні.

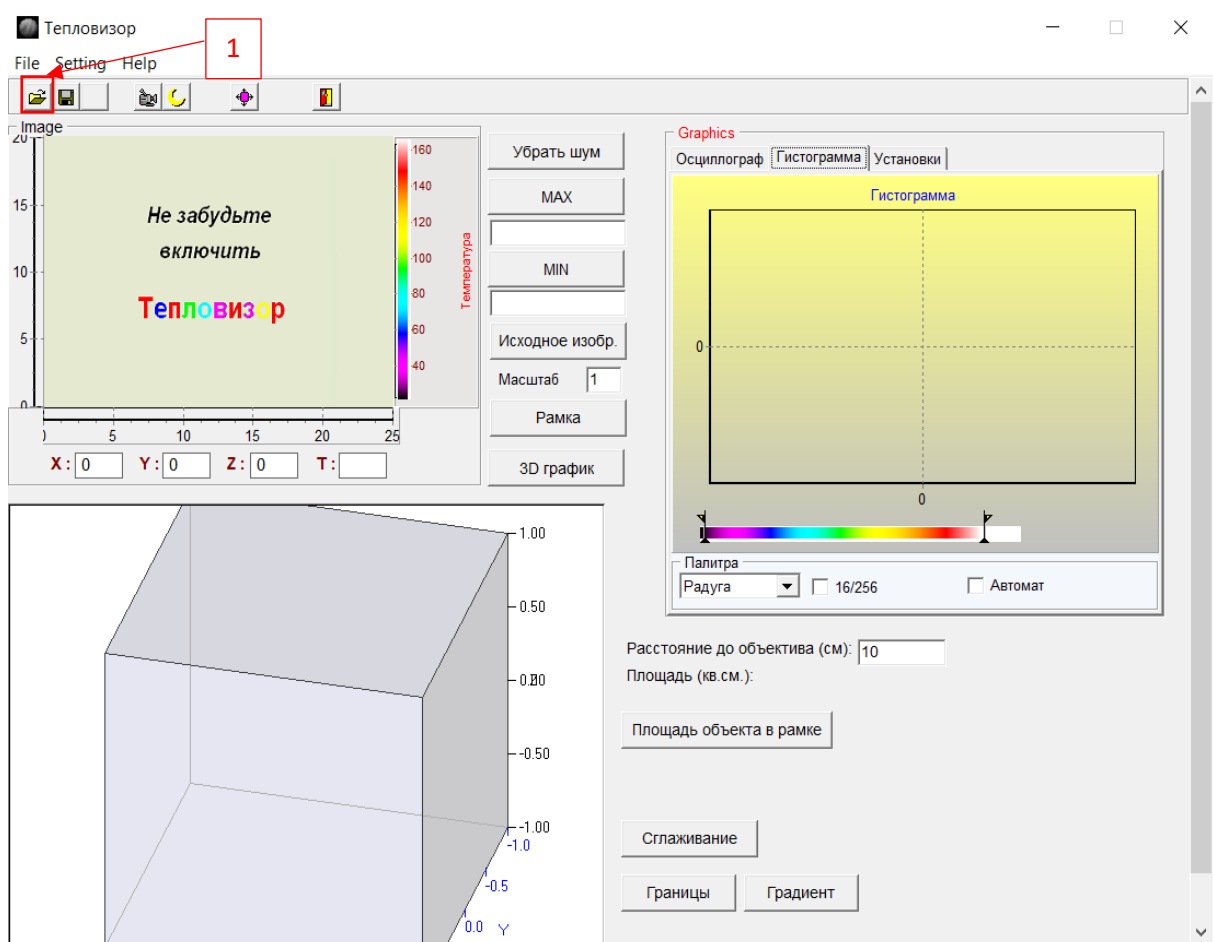


Рис.18. Вікно програми після її відкриття

2. Під час відкриття термограми запускається програмний код для усунення пошкоджених пікселів. Натиснути на зображення термограми в будь-якому місці. Під кнопками «MAX», «MIN» з'явиться значення максимальної та мінімальної температури (рис. 19). Натиснувши на ці кнопки, на термограмі залишаться лише області максимальної або мінімальної температури.

3. Натискаючи на кнопки з відповідною назвою відпрацьовують відповідні коди і на термограмі відображаються зміни, подвійну натиснення в будь-якій

точці термограми залишить температурний шар, який відповідає температурі в точці натиснення.

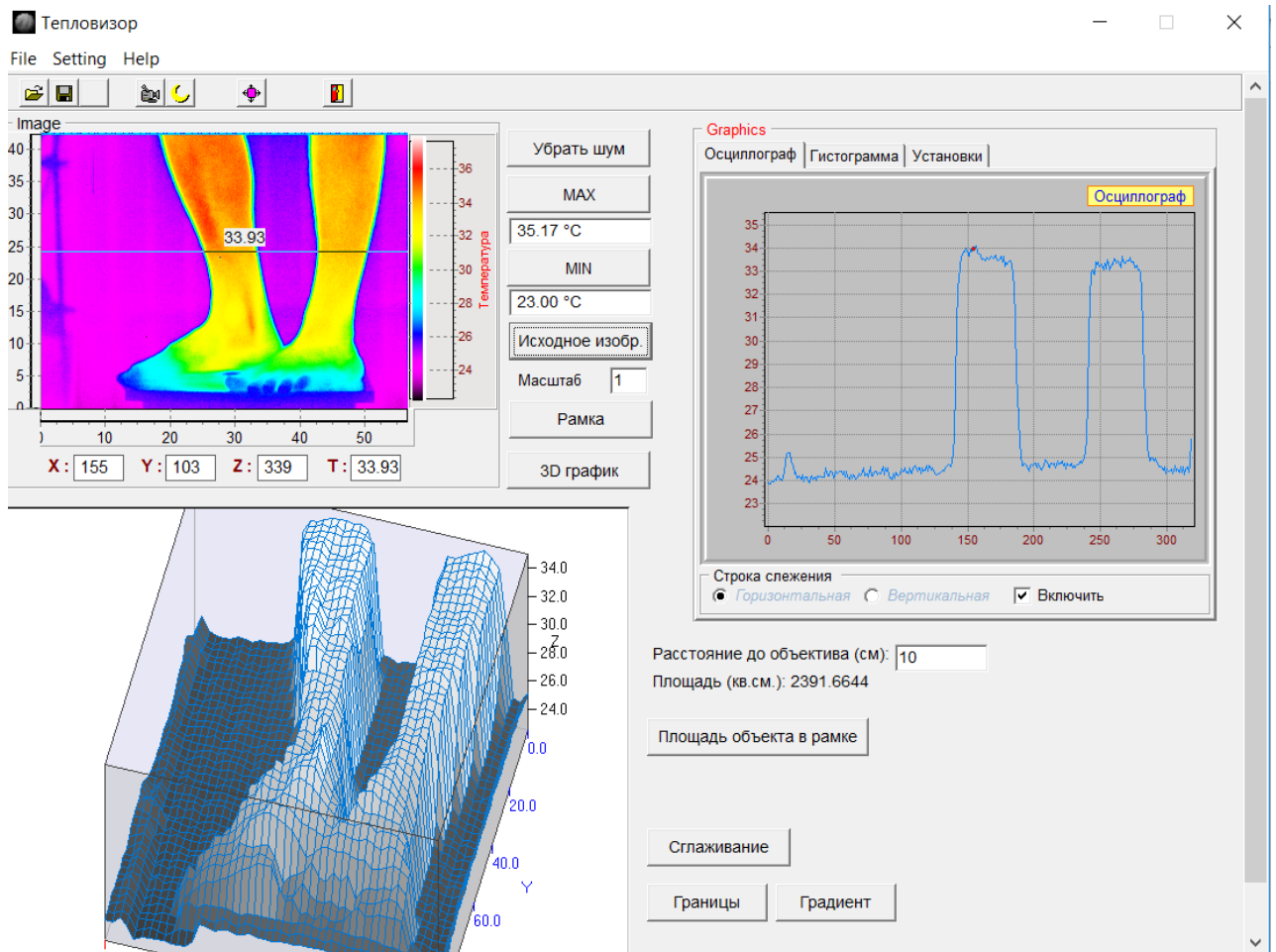


Рис.19. Вікно програми після відкриття термограми

4. До натискання на кнопку «Площадь объекта в рамке», розраховується вся площа термограми з ненульовими пікселями. Для кращого розуміння яка ділянка термограми використовується для визначення площі додана можливість виділяти цю область рамкою. Для зміни розміру рамки слід ввести коефіцієнт збільшення в поле «Масштаб», також частина термограми яка знаходиться в рамці масштабується і відображається на 3D графіку (рис. 20).

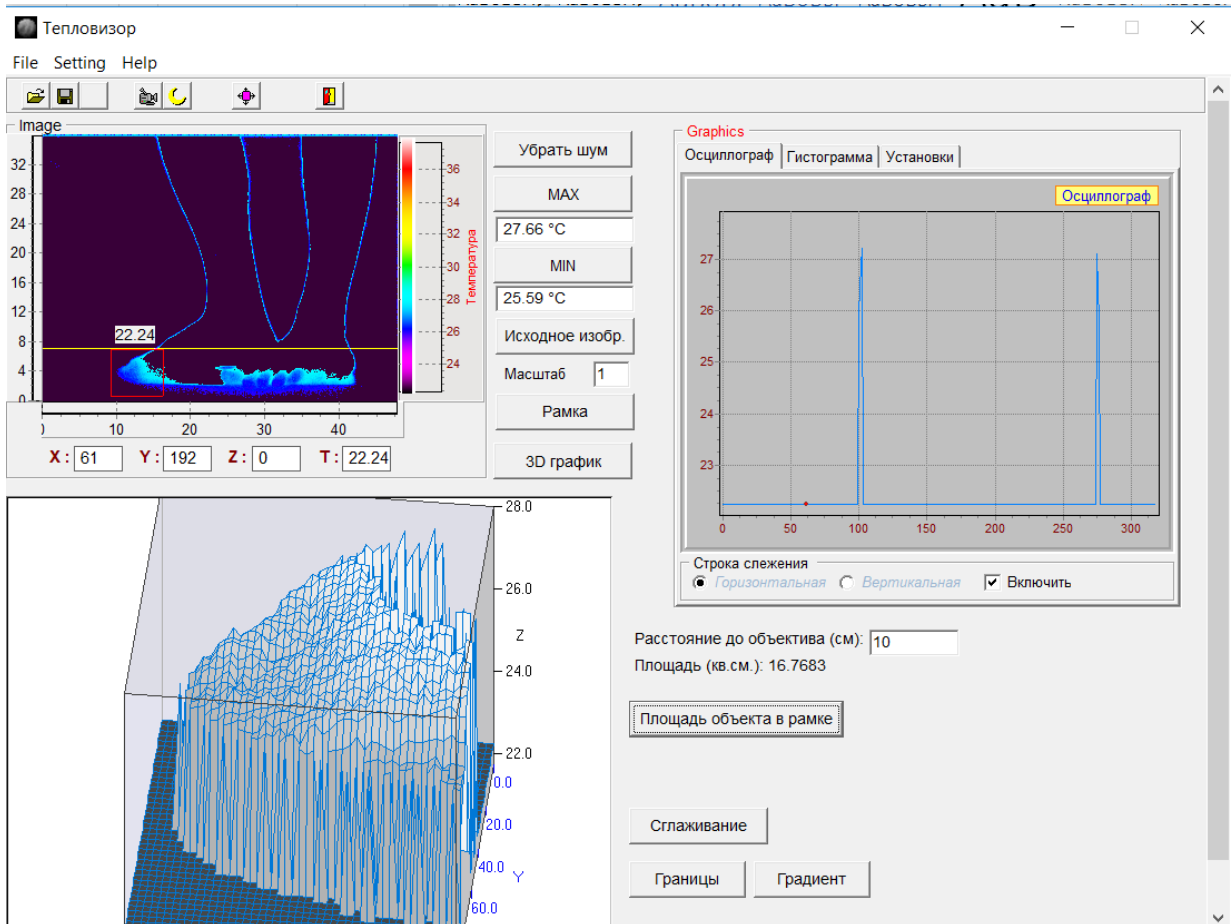


Рис. 20 Вікно програми після очищення термограми від шуму і виділення мінімальної області температури та масштабування певної області і визначення площі цієї області

## Висновки

1. В роботі показано, що шляхом вдосконалення існуючої програми можливо отримувати термографічні зображення з видаленими фоновими шумами та недосконалими пікселями, спрощувати отримання осцилографічних термозрізів в реальному значенні температур.

2. Розроблено спосіб визначення площі ураженої ділянки та отримання тривимірного зображення з можливістю вимірювати температуру по всій поверхні досліджуваного об'єкту, що значно розширює діагностичні можливості термографічних зображень.

3. Використання оновленого програмного забезпечення розширить діагностичні можливості термографів з метою більш якісної інтерпретації термографічних зображень.

## Література

1. Применение тепловидения в многопрофильных больницах и поликлиниках. Методические рекомендации // Под ред. д.т.н., проф. М.М. Мирошникова. Л., 1982. 84 с.
2. Вайнер Б.Г. Медицинское тепловидение высокого разрешения: новые возможности. // Врач. 1999. № 2. С. 25–27.
3. Зеновко Г.И. Термография в хирургии. М.: Медицина, 1998. 168 с.
4. Иваницкий Г.Р. Современное матричное тепловидение в биомедицине // Успехи физических наук. 2006. №176. С. 1293–1320.
5. Иваницкий Г.Р., Деев А.А., Хижняк Е.П., Хижняк Л.Н. Анализ теплового рельефа на теле человека // Технологии живых систем. 2007. Т. 4, №5-6. С. 43–50.
6. Иваницкий Г.Р., Деев А.А., Пашовкин Т.Н., Хижняк Е.П., Хижняк Л.Н., Цыганов М.А. Особенности теплового проявления подкожных источников нагрева на поверхности тела человека. // ДАН. 2008. Т. 420, № 4. С. 551–555.
7. Иваницкий Г.Р., Маевский Е.И., Смуров С.В., Хижняк Е.П., Хижняк Л.Н. Повышение диагностической информативности инфракрасных изображений с использованием методов нелинейного контрастирования // Известия института инженерной физики. 2016. №4 (42). С. 83–89.
8. Маевский Е.И., Смуров С.В., Хижняк Л.Н., Хижняк Е.П. Настоящее и будущее инфракрасной термографии // Известия института инженерной физики. 2015. №1. С. 2–12.
9. Хижняк Л.Н., Хижняк Е.П., Иваницкий Г.Р. Диагностические возможности матричной инфракрасной термографии. Проблемы и перспективы // Вестник новых медицинских технологий. 2012. № 4. С. 170–176.
10. Anbar M. Quantitative Dynamic Telethermograph in Medical Diagnosis. CRC Press: Boca Raton, 1994. P. 1–180.

11. Brian M. Sanchez, Mark Lesch, David Brammer, Susan E. Bove, Melissa Thiel, Kenneth S. Kilgore. Use of a portable thermal imaging unit as a rapid, quantitative method of evaluating inflammation and experimental arthritis // *Journal of Pharmacological and Toxicological Methods*. 2008. № 57. P. 169–175.
12. E. F. J. Ring and K. Ammer. Infrared thermal imaging in medicine. // *Physiological Measurement (IOP Publishing)*. 2012. № 33. P. 33–46.
13. Tay M.R., Low Y.L., Zhao X., Cook A.R., Lee V.J. Comparison of Infrared Thermal Detection Systems for mass fever screening in a tropical healthcare setting // *Public Health*. 2015. № 129. P. 1471–1478.
14. HyunJung Yang, Haeln Park, Chungsan Lim, SangKyun Park, KwangHo Lee. Infrared Thermal Imaging in Patients with Medial Collateral Ligament Injury of the Knee – A Retrospective Study // *Journal of Pharmacopuncture*. 2014. Vol. 17(4). P. 050–054.
15. Jenel Marian Patrascu, Mihaela Amarandei, Karla Noemy Kun, Qvidiu Boruga, Alina Totorean, Bogdan Andor, Sorin Florescu. Thermographic and microscopic evaluation of LARS knee ligament tearing // *Rom J. Morphol. Embryol*. 2014. Vol. 55 (3 Suppl). P. 1231–1235.
16. [https://www.testo.ru/ru-RU/funkcii\\_i\\_tekhnologii/po\\_testo\\_irsoft](https://www.testo.ru/ru-RU/funkcii_i_tekhnologii/po_testo_irsoft)
17. Кучерук І. М., Горбачук І. Т., Луцик П. П; за ред. Кучерука І. М. - К.: Техніка, 1999. Том 3: Оптика. Квантова фізика. - 520 с.
18. Приходченко В.В., Седаков И.Е., Приходченко О.В., Карначев А.С., Дорошев В.Д., Калинина О.И. Применение цифровой контактной термомаммографии в диагностике рака молочной железы // *Оригинальные Исследования*. 2007. Онкология Т 9 №2. С. 115–119.
19. Приходченко В.В., Приходченко О.В., Білошенко В.О., Карначов О.С., Дорошев В.Д., Калініна О.І. Можливості контактної цифрової термографії у ранній доклінічній діагностиці раку молочної залози // *Оригинальные Исследования*. 2011. Онкология Т 3 №2. С. 125–129.

20. Amatea R., Knoedler C. M., Pantuso F. P. et al. // Proceedings of SPIE, Infrared Technology and Applications XXIII. 1997. V. 3061. P. 210—222.
21. Серебренников П. С.: XVI Междунар. нач.-техн. конф. по фотоэлектронике и приборам ночного видения: Тез. докл. Ч. 5. Москва, 25—27 мая. 2000.
22. IR imager: one of the fastest growth applications in MEMS// MEMSentry, December 2007. No. 22. P. 9, 10.
23. Datskos P., Lavrik N.// SPIE Newsroom. 2006. DOI: 10.1117/2.1200608.0361.
24. Hunter S. R., Maurer G. S., Simelgor G., Radhakrishnam S., Gray J.// In Infrared Technology and Applications XXXIII/B. F. Andresen; G. F. Fulop, P. R. Norton, Editors, 65421F, 2007.
25. Lavrik N., Archibald R., Grbovic D., Rajic S., Datskos P.: Ibid.
26. Grbovic D., Lavrik N. V., Datskos P. G., Forrai D., Nelson E., Devitt J., McIntyre B.// Applied Physics Letters. 2006. V. 89. P. 073118-1.
27. Стафеев В.И., Болтарь К.О., Бурлаков И.Д., Акимов В.М., Кпиманов Е.А., Сагинов Л.Д., Соляков В.Н., Мансветов Н.Г., Пономаренко В.П., Тимофеев А.А., Филачёв А.М. Матричные фотоприемные устройства среднего и дальнего инфракрасных диапазонов спектра на основе фотодиодов из  $Cd_xHg_{1-x}Te$  // Государственный научный центр «НПО „Орион“» 2005
28. Гмурман В.Е. Теория вероятностей и математическая статистика. - М.: Высшая школа, 2003. - 479 с.
29. Фукунага К. Введение в статистическую теорию распознавания образов / К. Фукунага.- М.: Наука, 1979. - 368с.
30. Ray, Sidney F. Applied Photographic Optics: Lenses and Optical Systems for Photography, Film, Video // Electronic and Digital Imaging. Focal Press. 2002.- p. 40.

## Додаток А

Повний текст оновленого програмного коду:

```
unit UnitForTestImage;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, ComCtrls, Spin, Buttons, TeeProcs, TeEngine, Chart,
  Series, PaletteUnit, ToolWin, TeeFunci, Frame1Unit, Menus, jpeg,
  SDL_plot3d, Math;

type
  TForm1 = class(TForm)
    OpenDialog1: TOpenDialog;
    Timer1: TTimer;
    GroupBox2: TGroupBox;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    Chart2: TChart;
    Series4: TFastLineSeries; { Oscilloscope }
    Series5: TPointSeries; { point }
    Chart1: TChart;
    Image5: TImage;
    Image6: TImage;
    Series2: TFastLineSeries;
    Series3: TFastLineSeries;
    GroupBox1: TGroupBox;
    Label3: TLabel;
    Label4: TLabel;
    PosY: TEdit;
```

PosX: TEdit;  
PaletPanel: TImage;  
TeeFunction1: TLowTeeFunction;  
Chart3: TChart;  
Spectr2Panel: TImage;  
Series6: TLineSeries;  
MainMenu1: TMainMenu;  
File1: TMenuItem;  
Load1: TMenuItem;  
N1: TMenuItem;  
SaveImage: TMenuItem;  
SaveBitmap: TMenuItem;  
N3: TMenuItem;  
Exit1: TMenuItem;  
Setting1: TMenuItem;  
N4: TMenuItem;  
Help1: TMenuItem;  
N5: TMenuItem;  
About1: TMenuItem;  
TabSheet3: TTabSheet;  
ToolBar1: TToolBar;  
ToolButton2: TToolButton;  
SpeedButton1: TSpeedButton;  
SpeedButton2: TSpeedButton;  
SpeedButton3: TSpeedButton;  
ToolButton1: TToolButton;  
SpeedButton4: TSpeedButton;  
StatusBar1: TStatusBar;  
ToolButton3: TToolButton;  
SpeedButton5: TSpeedButton;

SpeedButton6: TSpeedButton;  
Timer2: TTimer;  
ToolButton4: TToolButton;  
SpeedButton7: TSpeedButton;  
GroupBox5: TGroupBox;  
HistoCheck: TCheckBox;  
ScopeCheck: TCheckBox;  
ComboBox2: TComboBox;  
CheckBox2: TCheckBox;  
Series1: TBarSeries;  
SaveDialog1: TSaveDialog;  
GroupBox3: TGroupBox;  
RadioButton1: TRadioButton;  
RadioButton2: TRadioButton;  
CheckBox1: TCheckBox;  
GroupBox4: TGroupBox;  
ComboBox1: TComboBox;  
AutoCheck: TCheckBox;  
Palette\_16: TCheckBox;  
Label1: TLabel;  
Label2: TLabel;  
Edit4: TEdit;  
Edit5: TEdit;  
Button1: TButton;  
Label6: TLabel;  
PosT: TEdit;  
SpinEdit4: TSpinEdit;  
TrackBar2: TTrackBar;  
SpeedButton8: TSpeedButton;  
SpeedButton9: TSpeedButton;

```
Edit_max_temp: TEdit;  
Edit_min_temp: TEdit;  
SpeedButton10: TSpeedButton;  
P3D: TPlot3D;  
SpeedButton11: TSpeedButton;  
Label_area: TLabel;  
Main_Image: TImage;  
Label_temp: TLabel;  
Edit_scale: TEdit;  
Label_Scale: TLabel;  
Label_frame1: TLabel;  
Label_frame2: TLabel;  
Label_frame3: TLabel;  
Label_frame4: TLabel;  
Button_Frame: TButton;  
Label5: TLabel;  
PosZ: TEdit;  
Chart4: TChart;  
Series7: TLineSeries;  
Edit1: TEdit;  
Label7: TLabel;  
Button2: TButton;  
Label10: TLabel;  
Edit8: TEdit;  
Label11: TLabel;  
Button3: TButton;  
Button4: TButton;  
Button5: TButton;  
Button6: TButton;  
Button7: TButton;
```

```
Chart5: TChart;
Series8: TLineSeries;
procedure DrawSpectr;
procedure PutFrame(Sender: TObject);
procedure PaintFrame(Sender: TObject);

procedure Main_ImageMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Init(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure Main_ImageMouseDown(Sender: TObject; Button:
TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Main_ImageMouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Image6MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure Image5MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure PaletPanelMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure ComboBox1Change(Sender: TObject);
procedure Exit1Click(Sender: TObject);
procedure Load1Click(Sender: TObject);
procedure Chart1Scroll(Sender: TObject);
procedure SpeedButton6Click(Sender: TObject);
procedure SpeedButton5Click(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure SpeedButton7Click(Sender: TObject);
procedure ComboBox2Change(Sender: TObject);
```

```
procedure CheckBox1Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure SaveImageClick(Sender: TObject);
procedure SaveBitmapClick(Sender: TObject);
procedure AutoCheckClick(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure SpinEdit2Change(Sender: TObject);
procedure TrackBar2Change(Sender: TObject);
procedure SpeedButton8Click(Sender: TObject);
procedure SpeedButton9Click(Sender: TObject);
procedure SpeedButton10Click(Sender: TObject);
procedure SpeedButton11Click(Sender: TObject);
procedure Button_FrameClick(Sender: TObject);
procedure FormDb1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Main_ImageDb1Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

const Timer_1_val : array[0..5] of integer = (40,50,66,100,200,500);

var
```

```
Form1: TForm1;  
copyImage: array [0..290,0..320] of integer;
```

```
dat : record  
    header: THeaderArray;  
    d : TFrameArray;  
end;
```

```
dat2: record  
    header: THeaderArray;  
    d : TFrameArray;  
end;
```

```
hair : record  
    x0, y0 : integer;  
    hor, vert : boolean;  
end;
```

```
dati : record  
    header: THeaderArray;  
    d : TDarkArray;  
end;
```

```
plot8 : array[0..4095] of integer;
```

```
b, b2, spectr1, spectr2 : TBitMap;
```

```
zmpos: byte;
```

```
PaletteGood, MoveMouse: boolean;
```

```
var tmp : TShiftState;
```

```
    y0, x0: integer;
```

```
MyPal : TMyPalette;
```

```
MyDat : TMyDatas;
```

```
MyFrame : TMyFrame;
```

```
MovePalet : boolean;
```

```
PaletPos : integer;
```

```
minp, maxp : integer;
```

```
T30, T35 : integer;
```

```
dt_k : double;
```

```
implementation
```

```
uses BigFormUnit;
```

```
{ $R *.DFM }
```

```
{ ===== }
```

```
===== }
```

START

```
Procedure TForm1.DrawSpectr;
```

```
var x, y : integer;
```

```
    k : longint;
```

```

    p : PARGB;
begin
  for y:=1 to 10 do
    begin
      p:= spectr1.scanline[y];
      for x:=0 to 255 do
        begin
          k:=x*16;
          p[x].r:= MyPal.UsedPalette.r[k];
          p[x].g:= MyPal.UsedPalette.g[k];
          p[x].b:= MyPal.UsedPalette.b[k];
        end;
      end;
    PaletPanel.canvas.draw(0,0,spectr1);

    for y:=0 to Spectr2Panel.Height-1 do
      begin
        p:= spectr2.scanline[y];
        k:= Round((Spectr2Panel.Height-1 - y)*$1000/Spectr2Panel.Height);
        if MyPal.UsedPalette.Pal_16 then
          k:=k and $F00 + $80;

          for x:=1 to 9 do
            begin
              p[x].r:= MyPal.MainPalette.r[k];
              p[x].g:= MyPal.MainPalette.g[k];
              p[x].b:= MyPal.MainPalette.b[k];
            end;
          end;
        end;
      Spectr2Panel.canvas.draw(0,0,spectr2);

```

```
Series6.Clear;
```

```
With Chart3.RightAxis do
```

```
begin
```

```
Automatic := False;
```

```
AutomaticMaximum := False;
```

```
AutomaticMinimum := False;
```

```
Minimum := (30+(MyPal.UsedPalette.min-T30)/dt_k);
```

```
if (35+(MyPal.UsedPalette.max-T35)/dt_k) > Minimum then
```

```
    Maximum := (35+(MyPal.UsedPalette.max-T35)/dt_k)
```

```
else Maximum := Minimum+20;
```

```
end;
```

```
Image6.Left:=(MyPal.UsedPalette.max div 16)+26;
```

```
Image5.Left:=(MyPal.UsedPalette.min div 16)+23;
```

```
end;
```

```
procedure Area();
```

```
var area, i, j :integer;
```

```
    Focus, distToObj, sizeFrameH,sizeFrameV,SquarePixel :real;
```

```
begin
```

```
area := 0;
```

```
for i:=0 to 239 do
```

```
    for j:=0 to 319 do
```

```
        if dat.d[i,j]> 0 then inc(area);
```

```
Focus := StrToFloat(Form1.Edit1.Text);
```

```
distToObj := StrToFloat(Form1.Edit8.Text);
```

```
if Focus <> 0 then
```

```
begin
    sizeFrameH := 320*(distToObj - Focus)/Focus;
    sizeFrameV := 240*(distToObj - Focus)/Focus;
    SquarePixel := sizeFrameV*sizeFrameH/76800;
end
else
begin
    sizeFrameH := 0;
    sizeFrameV := 0;
end;
```

With Form1.Chart4.BottomAxis do

```
begin
    Automatic := False;
    AutomaticMaximum := False;
    AutomaticMinimum := False;
    Minimum := 0;

    if sizeFrameH > Minimum then Maximum := sizeFrameH
    else Maximum := Minimum;
    Increment := trunc(Maximum*0.1);
end;
```

With Form1.Chart5.LeftAxis do

```
begin
    Automatic := False;
    AutomaticMaximum := False;
    AutomaticMinimum := False;
    Minimum := 0;
```

```

if sizeFrameV > Minimum then Maximum := sizeFrameV
  else Maximum := Minimum;
Increment := Form1.Chart4.BottomAxis.Increment;
end;

Form1.label11.Caption:= 'Площадь (px): ' + floattostr(area);
if Focus <> 0 then
  Form1.label_area.Caption:= 'Площадь (кв.см.): ' +
    floattostr(SimpleRoundTo(SquarePixel * area,-4))
  else Form1.label_area.Caption:= 'Площадь (кв.см.): infinity';
end;

procedure TForm1.PutFrame(Sender: TObject);
var x, y, z2 : integer;
    z :real;
    p, p_sub : PARGB;
    for_shl : byte;
begin

for y:=0 to 239 do
begin
p:= b.scanline[y];
p_sub := b2.scanline[y];

if (y=hair.y0) and hair.hor then
begin
for x:=0 to 319 do
begin
z:=dat.d[y,x];
p[x].r:= MyPal.UsedPalette.r[Round(z)] xor $FF;

```

```

    p[x].g:= MyPal.UsedPalette.g[Round(z)] xor $FF;
    p[x].b:= MyPal.UsedPalette.b[Round(z)];
end;
continue;
end;

for x:=0 to 319 do
begin
    z:=dat.d[y,x]+8;
    p[x].r:= MyPal.UsedPalette.r[Round(z)];
    p[x].g:= MyPal.UsedPalette.g[Round(z)];
    p[x].b:= MyPal.UsedPalette.b[Round(z)];

    if hair.vert and (x=hair.x0) then
    begin
        p[x].r:= MyPal.UsedPalette.r[Round(z)] xor $FF;
        p[x].g:= MyPal.UsedPalette.g[Round(z)] xor $FF;
        p[x].b:= MyPal.UsedPalette.b[Round(z)];
    end;
end;
end;

Main_Image.canvas.draw(0,0,b);
end;

procedure plot3d(startv, starth: real; x,y:integer);
var i,k:integer ;
new_temp,max,min :real;
begin;

```

```

form1.P3d.SuppressPaint := true;
form1.P3d.GridMat.Resize(47,42); // max size of matrix : 2000 elements

max:=0;
min:=1000;

for i:=1 to 47 do
begin;
  for k:=1 to 42 do
  begin;
    new_temp:= (30+(dat.d[trunc(k*startv+y),trunc(i*starth+x)]-T30)/dt_k);
    if (new_temp > max) then max:=new_temp;
    if (new_temp < min) and (new_temp<>0) then min:=new_temp;
    form1.P3d.GridMat[i,k] := new_temp;
  end;
end;

max:=Trunc(max)+1;
form1.P3D.ColorScaleHigh:=max;
if min < 28 then
  form1.P3D.SetRange(0,100,0,100,min-0.5,max)
else if (min >= 28) then
  form1.P3D.SetRange(0,100,0,100,min-4,max);
form1.P3d.SuppressPaint := false;
end;

procedure TForm1.PaintFrame(Sender: TObject);
var  q : boolean;
     x, y, z : integer;

```

```

begin
  q:=MyFrame.GrabFrame;

  if HistoCheck.Checked then
    begin
      FillChar(plot8, SizeOf(plot8),0);
      for y:=0 to 239 do
        for x:=0 to 319 do
          begin
            z:=MyFrame.frame.dat[y,x];
            z:=z-MyFrame.temn.d[y,x];
            if z<0 then z:=0;
            dat.d[y,x]:= z;
            inc(plot8[z]);
          end;

          minp:=0;
          maxp:=4095;
          for x:=1 to 4095 do if plot8[x]=0 then minp:=x else break;
          for x:=4095 downto minp+1 do if plot8[x]=0 then maxp:=x else break;
          if maxp<4084 then maxp:=maxp+10;
          if minp>10 then minp:=minp-10;
          AutoCheckClick(Sender);

          Series1.Clear;
          for x:=0 to 4095 do
            begin
              Series1.Add(plot8[x]/7.4,",clTeeColor);
            end;
          end { histogramme }

```

```

else for y:=0 to 239 do  {without hostogramme }
    for x:=0 to 319 do
        dat.d[y,x]:=MyFrame.frame.dat[y,x]-MyFrame.temn.d[y,x];

    PutFrame(Sender);
end;

procedure TForm1.Main_ImageMouseMove(Sender: TObject; Shift:
TShiftState; X,
    Y: Integer);
var x1, min, min2, max, i, j,area,Top_koef : integer;
    k, Scale, distToObj, Focus, sizeFrameH, sizeFrameV,SquarePixel :real;

begin
    if not MoveMouse then exit;
    if y>239 then y:=239;
    if y<0 then y:=0;
    if x>319 then x:=319;
    if x<0 then x:=0;

    PosX.Text:=' '+IntToStr(X);
    PosY.Text:=' '+IntToStr(Y);
    PosZ.Text:=' '+IntToStr(dat.d[y,x]);

    hair.y0:=y;
    hair.x0:=x;
    Scale := strtofloat(Form1.Edit_scale.Text);
    plot3d(Scale,Scale,x,y);

```

```

min:=dat.d[0,0];
max:=dat.d[0,0];
for i:=0 to 239 do
  for j:=0 to 319 do
    begin
      if min > dat.d[i,j] then min:= dat.d[i,j];
      if max < dat.d[i,j] then max:= dat.d[i,j];
    end;
  end;
end;

if ((30+(max-T30)/dt_k)*1.01) <= ((30+(min-T30)/dt_k)*0.99) then
  begin
    Chart2.LeftAxis.Maximum:= (30+(max-T30)/dt_k);
    Chart2.LeftAxis.Minimum:= (30+(min-T30)/dt_k);
  end
else
  begin
    Chart2.LeftAxis.Maximum:= (30+(max-T30)/dt_k)*1.01;
    Chart2.LeftAxis.Minimum:= (30+(min-T30)/dt_k)*0.99;
  end;
end;

Series4.Clear;

if hair.hor then
  for x1:=1 to 319 do
    begin
      k:= (30+(dat.d[y,x1]-T30)/dt_k);
      Series4.Add(k,",clTeeColor);
    end;
  end;

if hair.vert then

```

```
for x1:=1 to 239 do
```

```
begin
```

```
  k:= (30+(dat.d[x1,x]-T30)/dt_k);
```

```
  Series4.Add(k,"clTeeColor);
```

```
end;
```

```
Series5.Clear;
```

```
k:= (30+(dat.d[y,x]-T30)/dt_k);
```

```
if hair.hor then
```

```
  Series5.AddXY(x,k,"clTeeColor)
```

```
else Series5.AddXY(y,k,"clTeeColor);
```

```
min2:=max;
```

```
area := 0;
```

```
for i:=0 to 239 do
```

```
  for j:=0 to 319 do
```

```
    begin
```

```
      if (min2 > dat.d[i,j]) and (dat.d[i,j] <> 0) then min2:= dat.d[i,j];
```

```
      if dat.d[i,j]> min then inc(area);
```

```
    end;
```

```
Focus := StrToFloat(Edit1.Text);
```

```
distToObj := StrToFloat(Edit8.Text);
```

```
if Focus <> 0 then
```

```
begin
```

```
  sizeFrameH := 320*(distToObj - Focus)/Focus;
```

```
  sizeFrameV := 240*(distToObj - Focus)/Focus;
```

```
  SquarePixel := sizeFrameV*sizeFrameH/76800;
```

```
end  
else  
begin  
    sizeFrameH := 0;  
    sizeFrameV := 0;  
end;
```

```
With Form1.Chart4.BottomAxis do
```

```
begin
```

```
    Automatic := False;  
    AutomaticMaximum := False;  
    AutomaticMinimum := False;  
    Minimum := 0;
```

```
    if sizeFrameH > Minimum then Maximum := sizeFrameH  
    else Maximum := Minimum;  
    Increment := trunc(Maximum*0.1);
```

```
end;
```

```
With Form1.Chart5.LeftAxis do
```

```
begin
```

```
    Automatic := False;  
    AutomaticMaximum := False;  
    AutomaticMinimum := False;  
    Minimum := 0;
```

```
    if sizeFrameV > Minimum then Maximum := sizeFrameV  
    else Maximum := Minimum;  
    Increment := Form1.Chart4.BottomAxis.Increment;
```

```
end;
```

```

label11.Caption:= 'Площадь (px): ' + floattostr(area);
if Focus <> 0 then
    label_area.Caption:= 'Площадь (кв.см.): ' +
        floattostr(SimpleRoundTo(SquarePixel*area,-4))
else label_area.Caption:= 'Площадь (кв.см.): infinity';

Edit_max_temp.Text:= ' '+FloatToStrF(30+(max-T30)/dt_k,ffFixed,4,2) + '
°C';
Edit_min_temp.Text:= ' '+FloatToStrF(30+(min2-T30)/dt_k,ffFixed,4,2) + '
°C';

PosT.Text:= ' '+FloatToStrF(k{30+(k-T30)/dt_k},ffFixed,4,2);

{ Label with temperature above the cursor }
form1.label_temp.caption:=FloatToStrF((30+(dat.d[y,x]-
T30)/dt_k),ffFixed,4,2); // label on image
x:=x + Form1.Chart5.Width;
form1.label_temp.left:=x+4;
form1.label_temp.top:=y-4;

{ Frame }
Top_koef:=17;
form1.label_frame1.left := x;
form1.label_frame1.Top := y+Top_koef;
form1.label_frame1.Width := trunc(47*Scale);
form1.label_frame1.Height := 1;

form1.label_frame2.left := x;
form1.label_frame2.Top := trunc(y+42*Scale+Top_koef);

```

```

form1.label_frame2.Width := trunc(47*Scale);
form1.label_frame2.Height := 1;

form1.label_frame3.left := x;
form1.label_frame3.Top := y+Top_koef;
form1.label_frame3.Width := 1;
form1.label_frame3.Height := trunc(42*Scale);

form1.label_frame4.left := trunc(x+47*Scale);
form1.label_frame4.Top := y+Top_koef;
form1.label_frame4.Width := 1;
form1.label_frame4.Height := trunc(42*Scale);
end;

{=====}

procedure TForm1.Init(Sender: TObject);
var t: integer;

begin
// GroupBox1.Width:=Xm+4;
PosX.Ctl3D:=False;
PosY.Ctl3D:=False;
PosZ.Ctl3D:=False;
PosT.Ctl3D:=False;
PosX.Text:=' 0';
PosY.Text:=' 0';
PosZ.Text:=' 0';
Edit_max_temp.Text:=' ';
Edit_min_temp.Text:=' ';

```

```
label_frame1.Visible:=false;
label_frame2.Visible:=false;
label_frame3.Visible:=false;
label_frame4.Visible:=false;
Move(dat,dat2,SizeOf(dat));

PaletteGood:=false;

T30:= 225;
T35:= 370;
dt_k:= (T35-T30)/(35-30);
// dt_k:=1;

MyPal.InitPalettes;
MyPal.MakeMainPalette(0);
MyPal.MakeUsedPalette(0,4095);

b:=TBitmap.Create;
b.pixelformat:=pf24bit;
b.width:=Main_Image.Width;
b.height:=Main_Image.Height;

b2:=TBitmap.Create;
b2.pixelformat:=pf24bit;
b2.width:=Main_Image.Width;
b2.height:=Main_Image.Height;

spectr1:=TBitmap.Create;
```

```
spectr1.pixelformat:=pf24bit;
spectr1.width:=PaletPanel.Width;
spectr1.height:=PaletPanel.Height;

spectr2:=TBitMap.Create;
spectr2.pixelformat:=pf24bit;
spectr2.width := Spectr2Panel.Width;
spectr2.height:= Spectr2Panel.Height;

DrawSpectr;

MoveMouse:=false;
MovePalet:= false;

ComboBox1.ItemIndex:=0;
ComboBox2.ItemIndex:=1;
hair.y0:=0;
hair.x0:=0;
hair.hor:=RadioButton1.Checked;
hair.vert:=RadioButton2.Checked;

t:=MyFrame.Init;

end;

procedure TForm1.FormDestroy(Sender: TObject);
begin
    b.free;
    b2.free;
```

```
spectr1.free;  
spectr2.free;  
MyFrame.Close;  
end;
```

```
procedure TForm1.Main_ImageMouseDown(Sender: TObject; Button:  
TMouseButton;  
Shift: TShiftState; X, Y: Integer);  
begin  
MoveMouse:=true;  
hair.y0:=y;  
hair.x0:=x;  
end;
```

```
procedure TForm1.Main_ImageMouseUp(Sender: TObject; Button:  
TMouseButton;  
Shift: TShiftState; X, Y: Integer);  
begin  
MoveMouse:=false;  
hair.y0:=y;  
hair.x0:=x;  
end;
```

```
procedure TForm1.PaletPanelMouseMove(Sender: TObject; Shift: TShiftState;  
X, Y: Integer);  
var k: integer;  
begin  
if not MovePalet then Exit;  
k:=x-PaletPos;
```

```
MyPal.MakeUsedPalette(MyPal.UsedPalette.min+k,MyPal.UsedPalette.max+k);
```

```
    DrawSpectr;
```

```
    PaletPos:=x;
```

```
end;
```

```
procedure TForm1.Image6MouseMove(Sender: TObject; Shift: TShiftState; X,
```

```
    Y: Integer);
```

```
var k : integer;
```

```
begin
```

```
    if Shift=[ssLeft] then
```

```
        begin
```

```
            if (Image6.Left+x-4)<=(Image5.Left+7) then exit;
```

```
            Image6.Left:=Image6.Left+x-4;
```

```
            if Image6.Left>281 then Image6.Left:=281;
```

```
            k:=Image6.Left-26;
```

```
                if k>319 then k:=319;
```

```
                MyPal.MakeUsedPalette(MyPal.UsedPalette.min,k*16);
```

```
                DrawSpectr;
```

```
                Series2.Clear;
```

```
                Series3.Clear;
```

```
        end;
```

```
end;
```

```
procedure TForm1.Image5MouseMove(Sender: TObject; Shift: TShiftState; X,
```

```
    Y: Integer);
```

```
var k, n : integer;
```

```
begin
```

```
if Shift=[ssLeft] then
begin
  if (Image5.Left+x-6)>=(Image6.Left-7) then exit;

  Image5.Left:=Image5.Left+x-6;

  if Image5.Left<23 then Image5.Left:=23;

  k:=Image5.Left-23;

  if k<0 then k:=0;
  MyPal.MakeUsedPalette(k*16,MyPal.UsedPalette.max);
  DrawSpectr;

  Series2.Clear;
  Series3.Clear;
end;
end;

procedure TForm1.ComboBox1Change(Sender: TObject);
begin
  MyPal.UsedPalette.Pal_16:=Palette_16.Checked;
  MyPal.MakeMainPalette(ComboBox1.ItemIndex);
  MyPal.MakeUsedPalette(MyPal.UsedPalette.min, MyPal.UsedPalette.max);
  DrawSpectr;
end;
```

```

procedure TForm1.Exit1Click(Sender: TObject);
begin
    Form1.Close;
end;

procedure visualNoise;
var min,max,i,k,i1,k1,range,n, avpoint, denom:integer ;
    average, dispersion,sum, stDeviation :real;

begin
    min:=dat.d[0,0];
    max:=dat.d[0,0];
    sum:= 0;
    n:= 0;
    for i:=0 to 239 do
        for k:=0 to 319 do
            begin
                if min > dat.d[i,k] then min:= dat.d[i,k];
                if max < dat.d[i,k] then max:= dat.d[i,k];
                sum:=sum + dat.d[i,k];
                inc(n);
            end;
        range:=max-min;
        average:= sum/n;
        //ShowMessage(IntToStr(min)+' '+FloatToStr(average)+' '+IntToStr(max)+'
'+FloatToStr(sum)+' '+IntToStr(n));
        sum:=0;
        for i:=0 to 239 do
            for k:=0 to 319 do
                begin

```

```

    sum:= sum + (dat.d[i,k]*dat.d[i,k] - average*average);
end;
dispersion:= sum / n;
stDeviation:= sqrt(dispersion);
for i:=0 to 239 do
  for k:=0 to 319 do
    begin
      if (dat.d[i,k]<(average - 3*stDeviation)) or (dat.d[i,k] > (average +
3*stDeviation)) then
        begin
          {===== average value around non-standart point =====}
          avpoint:=0;
          denom:=0;
          for i1:=i-1 to i+1 do
            for k1:=k-1 to k+1 do
              begin
                if (dat.d[i1,k1]>(average - 3*stDeviation)) and (dat.d[i1,k1] <
(average + 3*stDeviation))
                  and not((i1=i) and (k1=k)) then
                    begin
                      avpoint := avpoint + dat.d[i1,k1];
                      inc(denom);
                    end;
                end;
              end;
            if denom>0 then avpoint:=round(avpoint/denom);
            {=====}
            dat.d[i,k]:= avpoint;
          end;
        end;
      end;
    end;
  //ShowMessage(FloatToStr(30+(average-T30)/dt_k));

```

end;

```
procedure TForm1.Button2Click(Sender: TObject);
```

```
begin
```

```
    Area();
```

```
end;
```

```
procedure TForm1.Load1Click(Sender: TObject);
```

```
var F : file;
```

```
    i, k, m, ymin, ymax, max : integer;
```

```
    xmax , xmin, x,y ,counts : integer;
```

```
    Focus, distToObj, sizeFrame :real;
```

```
begin
```

```
    MoveMouse:=false;
```

```
    if OpenFileDialog1.Execute then
```

```
        begin
```

```
            Timer1.Enabled:=false;
```

```
            AssignFile(F, OpenFileDialog1.FileName);
```

```
            Reset(F, 8);
```

```
            BlockRead(F, MyDat, 21669, counts);
```

```
            CloseFile(F);
```

```
            for i:=0 to 4095 do plot8[i]:=0;
```

```
// for i:=0 to 319 do plot16[i]:=0;
```

```
            ymin:=0;    ymax:=0;
```

```
            xmin:=0;    xmax:=0;
```

```

for i:=0 to 239 do
  for k:=0 to 319 do
    dat.d[i,k]:=MyDat.Frame[i,k] ;

visualNoise;
//min:=dat.d[0,0];
max:=dat.d[0,0];
for i:=0 to 239 do
  for k:=0 to 319 do
    begin
      //if min > dat.d[i,k] then min:= dat.d[i,k];
      if max < dat.d[i,k] then max:= dat.d[i,k];
    end;
max := Round(30+(max-T30)/dt_k);
MyPal.MakeUsedPalette(0,max*23);
for i:=0 to 239 do
  for k:=0 to 319 do
    begin
      inc(plot8[dat.d[i,k]]);
      copyImage[i,k]:= dat.d[i,k];
    end;

// Chart1.BottomAxis.SetMinMax( 0, xmax+20 );

minp:=0;
maxp:=4095;
for x:=1 to 4095 do
  if plot8[x]=0 then minp:=x else break;
for x:=4095 downto minp+1 do
  if plot8[x]=0 then maxp:=x else break;

```

```

if maxp<4084 then maxp:=maxp+10;
if minp>10 then minp:=minp-10;
AutoCheckClick(Sender);

```

```

Series1.Clear;
for i:=0 to 4095 do
begin
Series1.Add(plot8[i]/742.4,",clTeeColor);
end;

Timer2.Enabled:=true;
SpeedButton6.Enabled:=true;
StatusBar1.SimpleText:=' '+OpenDialog1.FileName+' ';
end;
Area();
end;

```

```

procedure TForm1.Chart1Scroll(Sender: TObject);
begin
if Chart1.BottomAxis.Minimum<0 then Chart1.BottomAxis.Minimum:=0;
if Chart1.BottomAxis.Maximum>4095 then
Chart1.BottomAxis.Maximum:=4095;
end;

```

```

procedure TForm1.SpeedButton6Click(Sender: TObject);
begin
Timer1.Enabled:=SpeedButton6.Down;

```

```
// SpeedButton6.Enabled:=false;
end;

procedure TForm1.SpeedButton5Click(Sender: TObject);
var q : boolean;
begin
    q:=MyFrame.GetDarkFrame;
end;

procedure TForm1.Timer2Timer(Sender: TObject);
begin
    PutFrame(Sender);
end;

procedure TForm1.SpeedButton7Click(Sender: TObject);
begin
    Form2.Show;
end;

procedure TForm1.ComboBox2Change(Sender: TObject);
begin
    Timer1.Interval:= Timer_1_val[ComboBox2.ItemIndex];
end;

procedure TForm1.CheckBox1Click(Sender: TObject);
begin
    if CheckBox1.Checked then
        begin
            hair.hor:=RadioButton1.Checked;
```

```
        hair.vert:=RadioButton2.Checked;
    end
else
    begin
        hair.hor:=false;
        hair.vert:=false;
    end;
end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
    Timer1.Enabled:=false;
    Timer1.Enabled:=true;
end;

procedure TForm1.SaveImageClick(Sender: TObject);
var F : file;
    b : boolean;

begin
    b:=Timer1.Enabled;
    Timer1.Enabled:=false;
    // MoveMouse:=false;
    if SaveDialog1.Execute then
        begin
            AssignFile(F, SaveDialog1.FileName);
            Rewrite(F, 8);
            with dat.header do
                begin
                    image_type[0]:='T';
```

```
    image_type[1]:='M';
    image_type[2]:='G';
    image_type[3]:='3';
    format[2]:=PZS_1;
    format[3]:=PZS_h;
    end;
    BlockWrite(F, Dat, 19264);
    CloseFile(F);
end;

Timer1.Enabled:=b;
end;

procedure TForm1.SaveBitmapClick(Sender: TObject);
var SourceRect, DestRect: TRect;
    x, y, dx, dy : integer;
    b: boolean;
    Bitmap: TBitmap;
begin

    b:=CheckBox1.Checked;
    CheckBox1.Checked:=false;
    PutFrame(Sender);
    Application.ProcessMessages;
// Panel1.Hide;

    x:=444;
    y:=295;
    dx:=256;
    dy:=290;
```

```

Bitmap := TBitmap.Create;
Bitmap.Width:=444;
Bitmap.Height:=295;

SourceRect      :=      Rect(GroupBox1.Left+1,      GroupBox1.Top+14,
GroupBox1.Left+4+dx, GroupBox1.Top+16+dy );
DestRect := Rect(1,1,x-62,y);
Bitmap.Canvas.CopyRect(DestRect,Form1.Canvas,SourceRect);

x:=Chart3.Left+GroupBox1.Left;
y:=Chart3.Top+GroupBox1.Top+8;
dx:= Chart3.Width;
dy:= Chart3.Height-16;

SourceRect:=Rect(x,y, x+dx, y+dy);
DestRect := Rect( 380, 1, 380 + dx, dy-2);
Bitmap.Canvas.CopyRect(DestRect,Form1.Canvas, SourceRect);

if SaveDialog1.Execute then
begin
    Bitmap.SaveToFile(ChangeFileExt(SaveDialog1.FileName, '.BMP'));
    ShowMessage(ChangeFileExt(SaveDialog1.FileName, '.BMP') + ' is saved');
end;
Bitmap.Free;
CheckBox1.Checked:=b;
end;

procedure TForm1.AutoCheckClick(Sender: TObject);
begin
    if AutoCheck.Checked then

```

```
begin
  MyPal.MakeUsedPalette(minp,maxp);
  DrawSpectr;
end
else
begin
  // MyPal.MakeUsedPalette(0,4095);
  DrawSpectr;
end
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  T30:= StrToInt(Edit4.Text);
  T35:= StrToInt(Edit5.Text);
  dt_k:= (T35-T30)/(35-30);
end;

procedure TForm1.SpinEdit2Change(Sender: TObject);
begin
  DrawSpectr;
  Chart3.Refresh;
end;

procedure TForm1.TrackBar2Change(Sender: TObject);
var tmp : integer;
begin
  tmp:=TrackBar2.Position*4+200;
  Edit4.Text:=IntToStr(tmp);
  tmp:=tmp+ (SpinEdit4.Value div 2);
```

```

Edit5.Text:=IntToStr(tmp);
Button1Click(Sender);
end;

procedure MinMax(extreme : String);
var min,max,i,k,range:integer ;
percent:real;
initialValue: array [0..239,0..319] of integer;

begin
min:=10000{dat.d[0,0]};
max:=dat.d[0,0];
for i:=0 to 239 do
for k:=0 to 319 do
begin
if (min > dat.d[i,k]) and (dat.d[i,k] <> 0) then min:= dat.d[i,k];
if max < dat.d[i,k] then max:= dat.d[i,k];
initialValue[i,k]:= dat.d[i,k];
end;
range:=max-min;
percent:=0.2*range;

if extreme = 'max' then
for i:=0 to 239 do
for k:=0 to 319 do
begin
if (dat.d[i,k] < (max-percent)) then dat.d[i,k]:=0;
end;
if extreme = 'min' then
for i:=0 to 239 do

```

```

for k:=0 to 319 do
  begin
    if (dat.d[i,k] > (min+percent)) then dat.d[i,k]:=0;
  end;
end;

procedure TForm1.SpeedButton8Click(Sender: TObject);
begin
  MinMax('max');
  Area();
end;

procedure TForm1.SpeedButton9Click(Sender: TObject);
begin
  MinMax('min');
  Area();
end;

procedure TForm1.SpeedButton10Click(Sender: TObject);
var min, max, i,k,range:integer ;
    percent :real;
begin
  min:=10000;
  max:=dat.d[0,0];
  for i:=0 to 239 do
    for k:=0 to 319 do
      begin
        if (min > dat.d[i,k]) and (dat.d[i,k] <> 0) then min:= dat.d[i,k];
        if max < dat.d[i,k] then max:= dat.d[i,k];
      end;
    end;
  end;
end;

```

```
range:=max-min;
percent:=0.11*range;

min := min + Round(percent);
  for i:=0 to 239 do
    for k:=0 to 319 do
      begin
        if dat.d[i,k] < min then dat.d[i,k] := 0;
      end;
    Area();
  end;

procedure TForm1.SpeedButton11Click(Sender: TObject);
begin;
  Form1.label_frame1.Visible:=false;
  Form1.label_frame2.Visible:=false;
  Form1.label_frame3.Visible:=false;
  Form1.label_frame4.Visible:=false;
  plot3d(5.71,6.8,0,0);
end;

procedure TForm1.Button_FrameClick(Sender: TObject);
begin
  label_frame1.Visible:=true;
  label_frame2.Visible:=true;
  label_frame3.Visible:=true;
  label_frame4.Visible:=true;
end;

procedure TForm1.FormDb1Click(Sender: TObject);
```

```
begin
  { Hide frame }
  //Form1.label_temp.Visible:=false;
  Form1.label_frame1.Visible:=false;
  Form1.label_frame2.Visible:=false;
  Form1.label_frame3.Visible:=false;
  Form1.label_frame4.Visible:=false;
end;

procedure TForm1.Main_ImageDblClick(Sender: TObject);
  var i,j, temp:integer;
begin
  temp := strtoint(PosZ.Text);

  for i:=0 to 239 do
    for j:=0 to 319 do
      if (temp-30>dat.d[i,j]) or (dat.d[i,j]> temp+30) then dat.d[i,j] := 0;
    Area();
  end;

procedure TForm1.Button3Click(Sender: TObject);
  var i,j, temp:integer;
begin

  for i:=0 to 239 do
    for j:=0 to 319 do
      dat.d[i,j] := copyImage[i,j];
    Area();
  end;
```

```

procedure TForm1.Button4Click(Sender: TObject);
var avpoint, denom,i,i1,j,j1:integer;
    origin, smooth: array [0..241,0..321] of integer;
begin

    for i:=0 to 241 do
    for j:=0 to 321 do
    begin
        origin[i,j]:= 0;
        smooth[i,j]:= 0;
    end;

    for i:=1 to 240 do
    for j:=1 to 320 do
    begin
        origin[i,j]:= dat.d[i-1,j-1];
        smooth[i,j]:= dat.d[i-1,j-1];
    end;

    for i:=1 to 240 do
    for j:=1 to 320 do
    begin
        avpoint:=0;
        denom:=0;
        for i1:=i-1 to i+1 do
        for j1:=j-1 to j+1 do
        begin
            avpoint:= avpoint+ origin[i1,j1];
            {if smooth[i1,j1] > 0 then }
            inc(denom);
        end;
    end;
end;

```

```

        end;
    {if denom = 9 then}
    smooth[i,j] := round(avpoint/denom);
end;

for i:=1 to 240 do
    for j:=1 to 320 do
        dat.d[i-1,j-1]:=smooth[i,j];
end;

procedure TForm1.Button5Click(Sender: TObject);
    var xmin,xmax,ymin,ymax,i,j,area:integer;
        scale,focus,distToObj,SquarePixel,sizeFrameV,sizeFrameH:real;
begin
    scale:= strtofloat(Edit_Scale.Text);
    xmin:= strtoint(posX.Text)+1;
    xmax:= xmin + round(47*scale)-1;
    ymin:= strtoint(posY.Text)+1;
    ymax:= ymin + round(42*scale);

    Focus := StrToFloat(Form1.Edit1.Text);
    distToObj := StrToFloat(Form1.Edit8.Text);
    area:=0;

    for i:= ymin to ymax do
        for j:=xmin to xmax do
            begin
                if dat.d[i,j]<>0 then inc(area);
            end;
        end;
    end;
end;

```

```

Form1.label11.Caption:= 'Площадь (px): ' + floattostr(area);
if Focus <> 0 then
begin
    sizeFrameH := 320*(distToObj - Focus)/Focus;
    sizeFrameV := 240*(distToObj - Focus)/Focus;
    SquarePixel := sizeFrameV*sizeFrameH/76800;
    Form1.label_area.Caption:= 'Площадь (кв.см.): ' +
        floattostr(SimpleRoundTo(SquarePixel*area,-4));
end
else Form1.label_area.Caption:= 'Площадь (кв.см.): infinity';

end;

procedure TForm1.Button6Click(Sender: TObject);
var sum,i,i1,j,j1:integer;
    origin, smooth: array [0..241,0..321] of integer;
    mask:array [0..2,0..2] of integer;
begin

    mask[0,0]:= 0; mask[0,1]:=-1; mask[0,2]:= 0; //mask[0,3]:= 0; mask[0,4]:= 0;
    mask[1,0]:=-1; mask[1,1]:= 4; mask[1,2]:=-1; //mask[1,3]:= 0; mask[1,4]:= 0;
    mask[2,0]:= 0; mask[2,1]:=-1; mask[2,2]:= 0; //mask[2,3]:= 0; mask[2,4]:= 0;
    { mask[3,0]:= 0; mask[3,1]:= 0; mask[3,2]:= 0; mask[3,3]:= 0; mask[3,4]:= 0;
    mask[4,0]:= 0; mask[4,1]:= 0; mask[4,2]:= 0; mask[4,3]:= 0; mask[4,4]:= 0; }

    for i:=0 to 241 do
        for j:=0 to 321 do
            begin
                origin[i,j]:= 0;
                smooth[i,j]:= 0;
            end
        end
    end
end;

```

```

end;

for i:=1 to 240 do
  for j:=1 to 320 do
    begin
      origin[i,j]:= dat.d[i-1,j-1];
      smooth[i,j]:= dat.d[i-1,j-1];
    end;

for i:=1 to 240 do
  for j:=1 to 320 do
    begin
      sum:=0;
      for i1:=0 to 2 do
        for j1:=0 to 2 do
          begin
            sum := sum + origin[i-1+i1,j-1+j1]*mask[i1,j1];
          end;
        smooth[i,j] := sum;
        if smooth[i,j] <0 then smooth[i,j] :=0;
      end;

for i:=1 to 240 do
  for j:=1 to 320 do
    dat.d[i-1,j-1]:=smooth[i,j];
end;

procedure TForm1.Button7Click(Sender: TObject);
var i,j:integer;
    origin, smooth: array [0..241,0..321] of integer;

```

```
begin

for i:=0 to 241 do
  for j:=0 to 321 do
    begin
      origin[i,j]:= 0;
      smooth[i,j]:= 0;
    end;

for i:=1 to 240 do
  for j:=1 to 320 do
    begin
      origin[i,j]:= dat.d[i-1,j-1];
      smooth[i,j]:= dat.d[i-1,j-1];
    end;

for i:=1 to 240 do
  for j:=1 to 320 do
    begin
      smooth[i,j] := origin[i,j]-origin[i-1,j];
      if smooth[i,j]<0 then smooth[i,j]:=0;
    end;

for i:=1 to 240 do
  for j:=1 to 320 do
    dat.d[i-1,j-1]:=smooth[i,j];
end;

End.
```