

ЕРЕВАНСКИЙ ФИЗИЧЕСКИЙ ИНСТИТУТ

Научное сообщение ЕФИ-96 (74)

В.М.КРИЩЯН, К.А. АМРОЯН, Д.Ш.ЗАГРАБЯН,  
Т.Н.ШАПОШНИКОВА

ПРОГРАММА ДЕШИФРОВКИ ИНФОРМАЦИИ  
О РЕПЕРНЫХ МЕТКАХ.

Ереван 1974

УДК.539.1.07

Научное сообщение ЕФИ-96(74)

**В.М.КРИЩЯН, К.А.АМРОЯН, Д.Ш.ЗАГРАБЯН,  
Т.Н. ШАПОШНИКОВА**

**ПРОГРАММА ДЕШИФРОВКИ ИНФОРМАЦИИ О  
РЕПЕРНЫХ МЕТКАХ**

**В статье изложен алгоритм, позволяющий на совокупности точек, измеренных сканирующим автоматом, выделить центры реперных меток и его программное осуществление.**

**Ереванский физический институт  
Ереван 1974**

**(С) Ереванский физический институт.1974**

V. M. KRISHCHYAN, K.A. AMROYAN,  
D.SH. ZAGRABYAN, T.N. SHAPOSHNIKOVA

PROGRAM OF FIDUCIAL POINTS  
INFORMATION DECODING

An algorithm for the separation of fiducial marks from the set of scanner measured points and its program implementation is presented.

Yerevan Physics Institute

Yerevan 1974

Описываемая ниже программа составлена для обработки плёнок на сканирующем автомате с горизонталью-го спектрометра [1] с широкозазорными искровыми камерами, собранного и установленного на высокогорной станции "Арагац" Ереванского физического института.

Поскольку количество информации, заключенное в одном кадре, невелико, решено было для облегчения составления программ не делать предварительного просмотра, а сканировать все кадры подряд, и в процессе обработки браковать ненужные кадры. Так как при остановке плёнки по маркам Бреннера положение кадра относительно просмотрного окна неизменно, то для выделения координат реперов пользовались постоянной цифровой маской. Затем в отдельные массивы группировались координаты точек, находящиеся в области реперов и служебной информации. Вся остальная информация классифицировалась как трек. Дальше программно выделялась информация о треке и обрабатывалась.

На рис.1 приводится изображение, полученное на плёнке и подлежащее обмеру.

После обмера кадра на сканирующем автомате полученная информация разделяется на массивы. В 1 массиве указывается число реперов, количество точек на каждом репере и координаты всех точек, лежащих в области радиуса  $R$ , выделенной как местонахождение репера. По этим координатам и определяются координаты центра репера.

Углы  $\alpha_1$  и  $\alpha_2$ , которые составляют ветви репера с осями стола измерительного прибора, определяются предварительно с точностью 0,1 рад.

Из массива координат точек, относящихся к 1 реперу (массив  $T$ ), выбирается 1 точка  $(X_1, Y_1)$ , через неё проводится прямая под углом  $\alpha_1$  или  $\alpha_2$  к оси  $X$ , которая параллельна кромке плёнки.

Уравнение этой прямой:

$$y = Kx + A,$$

где  $K = \operatorname{tg} \alpha$

$$A = y_1 - Kx_1$$

Затем из массива  $T$  выбираются точки, отстоящие от этой прямой на расстоянии, например,  $R/10$  — эта величина определяется экспериментально.

Если это число точек меньше одной трети числа точек всего массива  $T$ , то выбирается вторая точка, через неё проводится прямая, отбираются точки и т.д., пока число отобранных точек будет больше  $1/3$  числа точек в массиве  $T$  (программа  $RHP\Phi$ ).

Тогда поэтём отобранным точкам по методу наименьших квадратов аппроксимируется прямая, вычисляется дисперсия точек от прямой (программа  $RAM1[2]$ ).

Браковка точек, не принадлежащих выбранной ветви репера, реализована в виде итерационного процесса. Отбрасываются точки, расстояние которых от новой прямой больше  $2\sigma$  для первых двух итераций, а затем больше  $3\sigma$ , где

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (y_i - Kx_i - a)^2}{n}}$$

—среднеквадратичное отклонение точек от аппроксимирующей прямой. По оставшимся точкам снова аппроксимируется прямая и т.д., пока при очередном повторении не будет отброшенных точек (программа  $LUB1$ ). Аналогичный процесс дает возможность определить и уравнение второй ветви репера и затем как координата центра репера берется точка пересечения этих двух прямых.

Процесс повторяется, пока не будут определены координаты центров всех измеренных реперов ) (программа RHPD).

Затем по соответствующей программе RPR определяется взаимное расстояние между реперами и сравнивается с эталоном. В этой же программе предусмотрена возможность внесения поправок в координаты реперов (третьего и четвертого), чтобы эти реперы были выставлены по одной линии на двух половинках установки, фотографируемых разными фотоаппаратами. Далее определяются параметры для перехода от системы координат, связанной с измерительным прибором, к системе координат, связанной с реперами.

Система координат, связанная с реперами, выбирается так, чтобы начало координат размещалось в точке расположения третьего репера, а ось X, совпадающая с направлением движения частиц, совпадала с линией, соединяющей третий и четвертый реперы.

Преобразование системы координат измерительного стола в систему, связанную с реперами, рассматривается как трансляция и поворот:

$$x_i' = (x_i - \Delta x) \cos \varphi + (y_i - \Delta y) \sin \varphi$$

$$y_i' = -(x_i - \Delta x) \sin \varphi + (y_i - \Delta y) \cos \varphi,$$

где  $\varphi$  - угол, составленный прямой, проходящей через третий и четвертый реперы с осью ординат стола.

Параметры этого преобразования вычисляются из следующих выражений:

$$\cos \varphi = \frac{y_4 - y_3}{\sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2}}$$

$$\sin \varphi = \frac{x_4 - x_3}{\sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2}}$$

$$\Delta x = x_{\text{III репера}}$$

$$\Delta y = y_{\text{III репера}}$$

• Определяется масштаб фотографирования:

$$\omega = \frac{A}{\sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2}}$$

Ниже приводится блок-схема программы, распечатки всех используемых в ней подпрограмм и принятые обозначения.

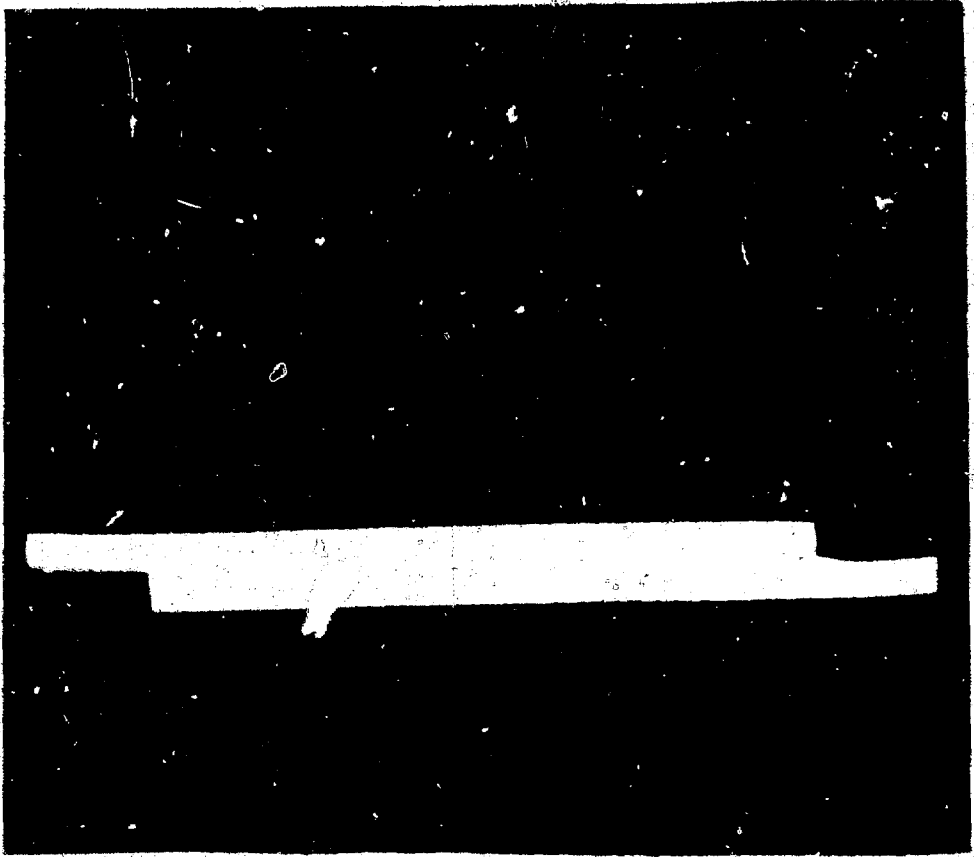
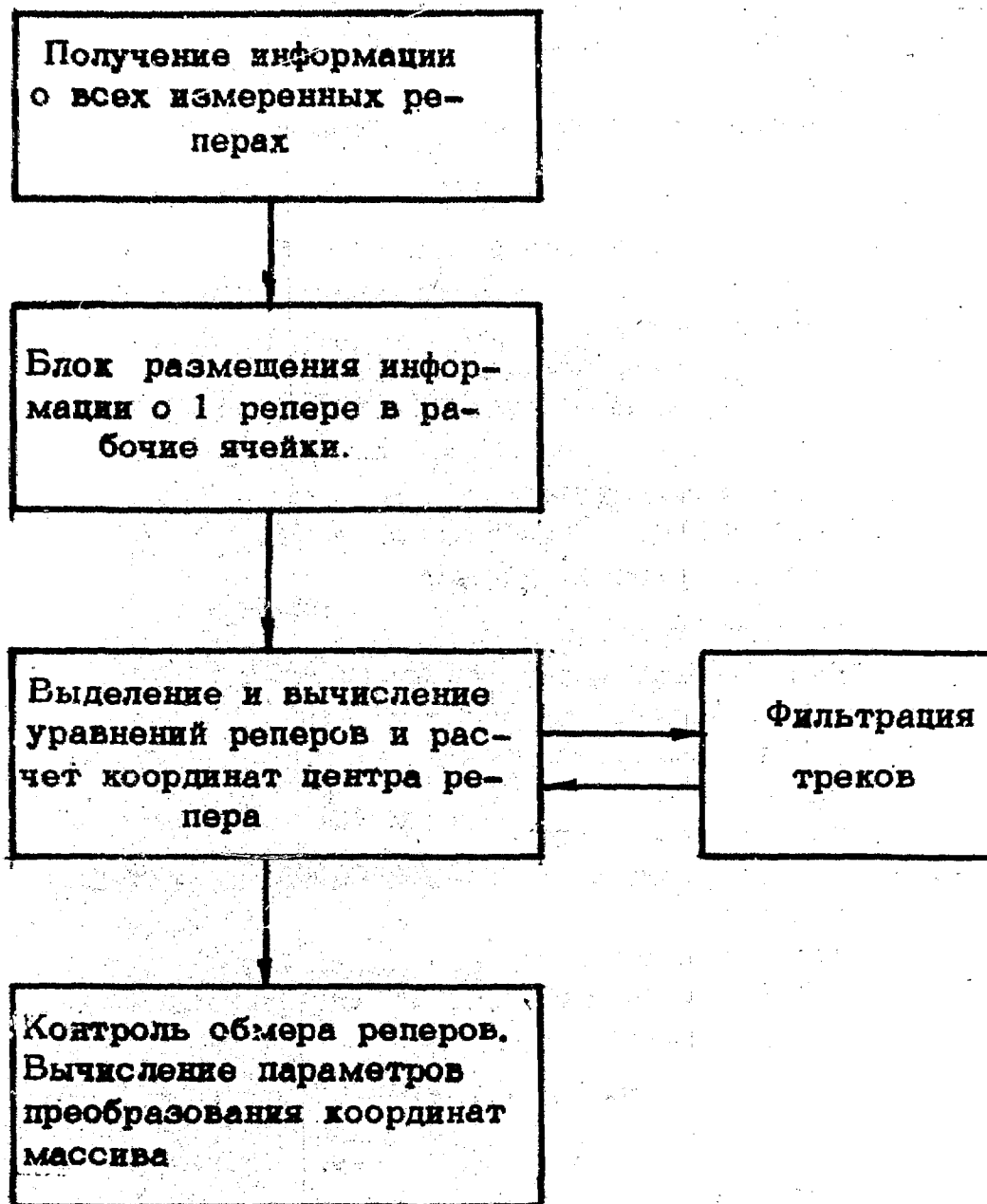


Рис.1

На снимке видны: штриховая информация; шесть крестообразных реперов; трек, состоящий из двух половинок и двух отрезков на его концах, появившихся в результате отклонения в клине, использованном для определения глубины прохождения трека; марка Бреннера.

## БЛОК - СХЕМА



## УСЛОВНЫЕ ОБОЗНАЧЕНИЯ

- IYU** - информация, полученная с измерительного прибора.  
**T** - массив координат точек, относящихся к области репера .  
**TM** - массивы координат точек отдельных ветвей репера.  
**IX, IY** - массивы, созданные для высвечивания на дисплее  
**X2X** - массив координат центров реперов.  
**D, DD** - расстояние от точки до аппроксимирующей прямой  
**KR** - количество реперов.  
**IIM** - индексация введенной информации.  
**II, K, KK, KAT, N, NN** - индексация перечисленных выше массивов.  
**NK** - порядковый номер репера.  
**A, UMV, NOM, NUM, NUMB** - количество точек, по которым аппроксимируется прямая.  
**R** - радиус области репера.  
**DELTA** - среднеквадратичный разброс точек от прямой.  
**H, A** - параметры уравнения прямой.  
**XC, YC** - координаты центра репера.  
**J** - перебор реперов.  
**TETA** - эталон точности угла.  
**KORA, KWANT** - служебные константы для модификации программы.  
**SIGD** - ячейки сумматоры.  
**SANG, DELH, DELA** - точность в определении параметров прямой.

ARRAY

```
C   PROGRAM ARRAY
      LOGICAL ISENSW
      DIMENSION NRP(10), IXR(10,30), IYR(10,30), FN1(2), T(60),
      D(20), TU(40), X2X(30)
      DATA FN1(1), FN1(2), /4HFILD, 4H SRC/
      CALL SEEK (3, FN1)
      READ (3) KR, NRP, IXR, IYR
      IF (ISENSW(1)) WRITE(4, 100) KR, NRP
100  FORMAT((10X, 2HKR, I10) / ((10X, 2HNRP) / ((10I7)))
      IF (ISENSW(0)) WRITE(4, 101) IXR, IYR
101  FORMAT((10X, 10HIXR, IYR) / ((10I7)))
      NK=1
      DO 2 IRI=1, KR
      NUM=NRP(IRI)
      L=1
      DO 1 I=1, NUM
      T(L)=FLOAT(IXR(IRI, I)) / 1000.
      T(L+1)=FLOAT(IYR(IRI, I)) / 1000.
1    L=L+2
      CALL RHPD(NUM, T, X2X, NK, L)
2    NK=NK+2
      CONTINUE
      STOP
      END
```

TANA

SUBROUTINE TANA (T, KAT)

```

LOGICAL ISENSW
DIMENSION IX(30), IY(30), T(60)
314 IT=0
KAT=KAT-1
RAT=T(2)
DO 313 I=4, KAT, 2
RAT=AMAX1(RAT, T(I))
313 CONTINUE
DO 311 I=1, KAT, 2
IT=IT+1
IX(IT)=IFIX((T(I)-T(1)) / 10000.)
311 IY(IT)=IFIX(RAT-T(I+1))*10000.)
CALL IDSPXY (IT, IX, IY)
CONTINUE
IF (ISENSW((0)) WRITE (4, 312) IX, IY, IT
312 FORMAT (1X10I5)
RETURN
END

```

RHPD

```

SUBROUTINE RHPD(NUM, T, X2X, NK, L)
LOGICAL ISENSW
DIMENSION T(60), TU(40), IX(30), IY(30), X2X(30)
R=0.03
CALL TANA (T, L)
IF (ISENSW(5)) WRITE(4, 307) NUM, T
H=1.
317 NH=1

```

```

X2X(NK+1)=YC
IF(ISENSW(5))WRITE(4,307)NK,X2X(NK),X2X(NK+1)
307  FORMAT(I5/(6F10.5))
      RETURN
301  TU(K)=T(I)
      TU(K+1)=T(I+1)
      K=K+2
      GO TO 300
      END

```

LUB1

```

SUBROUTINE LUB1(NUMB,T,DELTA,H,A,KORA,KWANT)
LOGICAL ISENSW
DIMENSION D(30),T(60)
NAT=0
15  CALL RAM1 (T,H,A,NUMB,1,D,DELTA,1)
      NAT=NAT+1
      IF(ISENSW(7))WRITE(4,10)NUMB,D
      IF(NAT.GT.1) GO TO 2
      DO 4 I=1,NUMB
      IF (D(I)-1000.)4,3, 3
4  CONTINUE
2  N=NUMB
      IF(NAT.LE.2) GO TO 47
      DO 40 I=1, NUMB
      IF(ABS(D(I))-3.*DELTA)40,40,42
40 CONTINUE
      IF (NUMB.LT.5) GO TO 6

```

```

304     KK=-1
302     KK=KK+2
        A=T(KK+1)-H*T(KK)
        IF(ISENSW(6))WRITE(4,307)KK,A
        K=1
        DO 300 I=1,1,2
        DD=(H*T(I)+A-T(I+1))/SQRT(1.+H**2)
        IF(ABS(D).LT.(R/10.)) GO TO 301
        CONTINUE
        IF(ISENSW(7))WRITE(4,307)I,DD
300     CONTINUE
        CALL TANA (TU,K)
        IF(ISENSW(7))WRITE (4,307)K,TU
        NUMB=K/2
318     IF((K/2).LT.(NUM/3)) GO TO 302
        CALL LUB1 (NUMB,TU,DELTA,H,A)
        IF(ABS(ABS(H)-1).GT.0.2)GO TO 302
        IF(NH.GE.2) GO TO 305
        H1=H
        A1=A
        H=-1
        NH=NH+1
        GO TO 304
305     H2=H
        A2=A
        XC=(A2-A1)/(H1-H2)
        YC=H1*XC+A1
        X2X(NK)=XC

```

```

IF(N.GT.NUMB) GO TO 15
IF(ISENSW(7))WRITE (4,12)NUMB,N,T
10.  FORMAT (I5/(6F10.5))
      IF (XORA.LE.1)CALL FILT1(T,H,A,NUMB,0.2)
      RETURN
11   FORMAT(2(10I5/)(10F5.3))
3    CALL FILT1(T,H,A,NUMB,0.02)  GO TO 15

12   FORMAT(2I10/(6F10.5))
47   DO 44 I=1,NUMB
      IF(ABS(D(I))-2.*DELTA) 44,44,42
44   CONTINUE
      GO TO 15
42   DO 43 K=I, NUMB
      T(2*K-1)=T(2*K+1)
      T(2*K)=T(2*K+2)
      D(K)=D(K+1)
      IF(ISENSW(7))WRITE(4,45)K,I
45   FORMAT (2I5)
43   CONTINUE
      NUMB=NUMB-1
      I=I-1.
      GO TO 40
46   PAUSE 46
      CONTINUE
      RETURN
      END

```

FILT 1

```
SUBROUTINE FILT1(T,H,A,NUMB,TETA)
LOGICAL ISENSW
DIMENSION Z(60), T(60)
NUMB=NUMB+1
35 J=1
NUMB=NUMB-1
IF( ISENSW(0) ) WRITE(4,37)H
TAN=H
DO 35 J=1, NUMB
CALL OTBR(T,J,Z,NUMB)
NOM=NUMB-1
CALL RAM1(Z,H,A,NOM,1,D,DELTA,1)
IF(ABS(TAN-H)-TETA) 30,33,33
30 CONTINUE
IF( ISENSW(0) ) WRITE(4,37) H
37 FORMAT(F10.5)
35 CONTINUE
RETURN
33 DO 34 K=J, NUMB
T(2*K-1)=T(2*K+1)
34 T(2*K)=T(2*K+2)
IF(J.LE.NUMB) GO TO 36
WRITE(4,38) J,NUMB
38 FORMAT(2I10,10H BAD )
STOP
END
```

RAM 1

SUBROUTINE RAM1(X,H,A,NUMB,K,D,DELTA,NG)

LOGICAL ISENSW

DOUBLE PRECISION DX,DY,SIGY2,SIGXY,SIGX,SIGX2,SIGY,SIGD,

\* UMB

REAL H,A,DELTA

DIMENSION X(60),D(30)

UMB=NUMB

SIGXY=SIGX-SIGX2-SIGY-SIGY2,SIGD=0

NMB=2\*NUMB+K-1

DO 2 I=K,NMB,2

DX=X(I)

DY=X(I+1)

SIGX=SIGX+DX

SIGX2=SIGX2+DX\*\*2

SIGXY=SIGXY+DX\*DY

SIGY=SIGY+DY

2 SIGY2=SIGY2+DY\*\*2

H=(UMB\*SIGXY-(SIGX\*SIGY))/(UMB\*SIGX2-SIGX\*\*2)

IF(ISENSW(0))WRITE(4,5)NUMB,SIGX,SIGX2,SIGXY,SIGY,SIGY2

A=(SIGX2\*SIGY-SIGX\*SIGXY)/(UMB\*SIGX2-SIGX\*SIGX)

L=1

DO 3 I=K,NMB,2

D(L)=H\*X(I)+A-X(I+1)

SIGD=SIGD+D(L)\*\*2

3 L=L+1

IF(ISENSW(1))WRITE(4,8)SIGD

```
3      FORMAT(P15.5/)
      DELTA=SQRT(SIGD/((1.+H**2)*UMB))
      IF(ISEMSW(1))WRITE (4,5)NUMB,DELTA,H,A
5      FORMAT(I3/(6P10.2))
      IF(NG.EQ.2) GO TO 4
      RETURN
      END
```

OTER

```
      SUBROUTINE OTER (T,J,Z,NUMB)
      DIMENSION T(60),Z(60)
      DO 31 K=1, NUMB
      IF(K.LT.J) GO TO 32
      Z(2*K-1)=T(2*K+1)
31      CONTINUE
      RETURN
32      Z(2*K)=T(2*K)
      Z(2*K-1)=T(2*K-1)
      GO TO 31
      END
```

## ЛИТЕРАТУРА

1. *Alchudjan S.V. et al. PICCR, Canada, 4, 1169*
2. Р.А.Фишер. Статистические методы для исследований. Государственное статистическое издательство, Москва, 1958 г.

Рукопись поступила 5-го сентября 1974г.



Редактор Л.П.Мукаян  
тех.редактор А.С.Абрамян

Заказ 0978

ВФ -03468

Тираж 300

---

Подписано к печати 31/ХП-74г. Формат издания 30х40  
1,0 уч.изд.л. Ц.7 к.

---

Отпечатано на ротапринтере  
Ереванского физического института, Ереван 36, ул.Мар-  
каряна 2