КОМПЬЮТЕРНАЯ МАТЕМАТИКА

В.А. ПЕТРУХИН, А.С. ДЖУС, С.Х. ШАЙХЛИСЛАМОВ,
Е.В. НЕЧЕПУРЕНКО

МОБИЛЬНАЯ ТЕХНОЛОГИЯ ИНТЕРАКТИВНОЙ ВИЗУАЛИЗАЦИИ ТРЕХМЕРНЫХ МОДЕЛЕЙ

.

,

(　　　-
)　　　　　　, . .　　　　　　　　　　　　　　.

. .　　　　　,　-

1982　　　　　　　　　　　　　　　　　　　,

,

[1].

,

.

-

,

.

1.

,　　　　),

(　　　　　　　　　　　　　　　),

.　　　　　　,

,　　　　　　　　　　　,

,

.

,

,　　　　　　　　　　　　　,

.

-

.

-

,

,

.

1.

.

2.

.

3.

,　　　　　　　　　　　　　.

4.　　　　　　　　　　　　　　　　　　　.
2.

VR-　　　　　　　,　　　　　　VR-

.　　　　　　　　　　　　　　　　　-

.

,   –

.   ,

.

,

.

,

.

–   .

0.25°   ,   .

130   ,   70   ,

.

TCP/IP   UDP [7, 8].

3.

,   ,

–

.

–   .   –

,

,   ,   .

1, 3   4,

4.

4.1.   .

–   .   ,

,

,

.   –

.

,

(   ).   ,

(. 1).

,   ,   ±30   ,

.

− 3   .

.   –

,

(   ).   –

,   .

(7   )   (1.5   )   . 2.

, -
.
. -

. 3.

, -
[2, 3].

. 1.

. 2.

. 3.

4.2. . , -
-
, .

. 40 .
,
(, ,
) . , ,
( ).
4.3. . -
, (-
,). -
-
.
«SvBundle» « - ». SvBundle
- , . ,
( ).
. 4. -
.

. 4.

4.4. .

. 5.

,   -
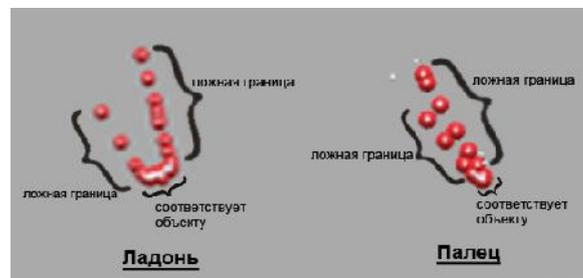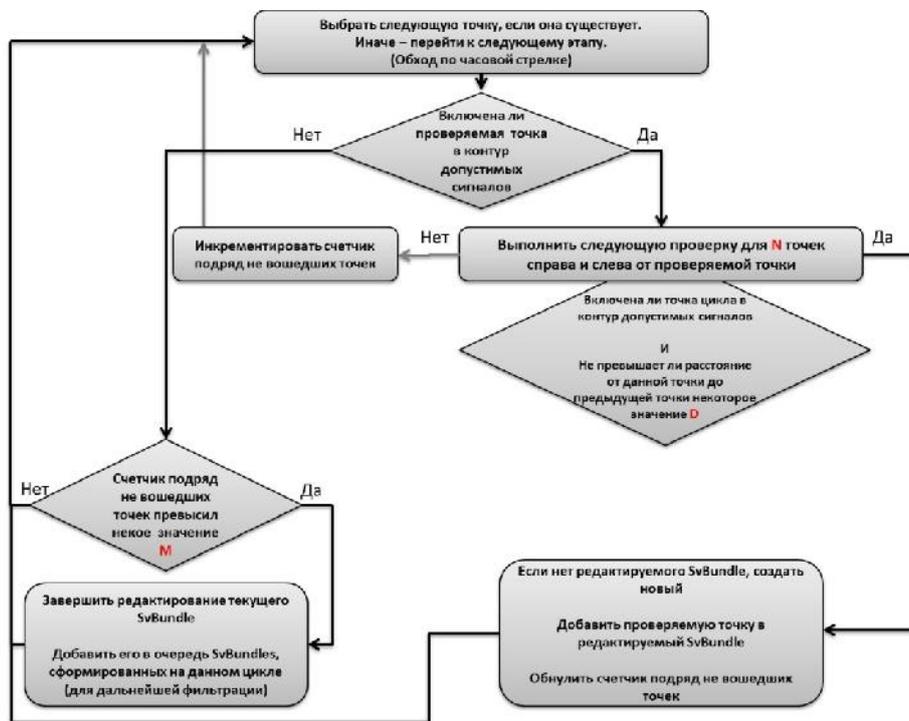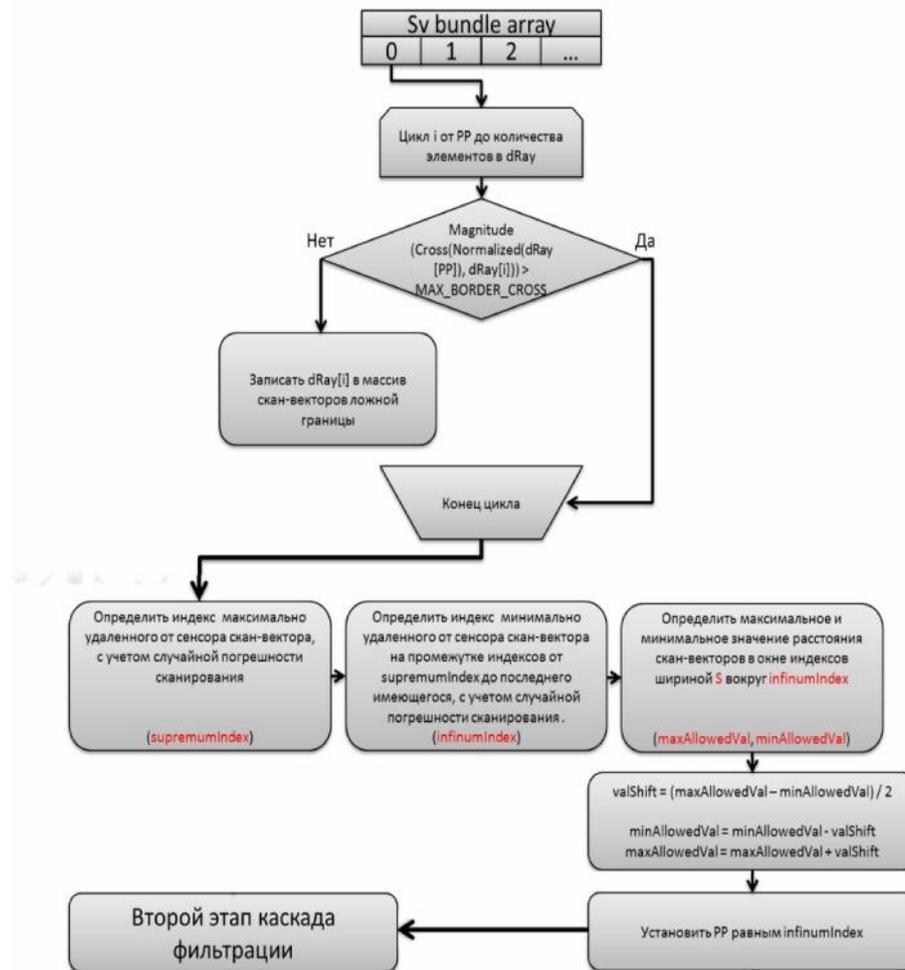   (
10   300   )   :

$Magnitude(Cross(Norm(dRay[PP]), dRay[i])) > MAX\_BORDER\_CROSS,$

dRay –   -
                                                                           ;
i – $PP + 1$ *dRay*
                                                    ;
Cross –   ;
Magnitude –   ;
Norm –   , *MAX_BORDER_CROSS*;
S –   .

                              -                        (*PP*,   . «progress pointer»,
).   progress pointer   -
                                                                          .

4.5.   .
                                                   -                    ,   -
              .
                   -   (. 2),

          (. 3).
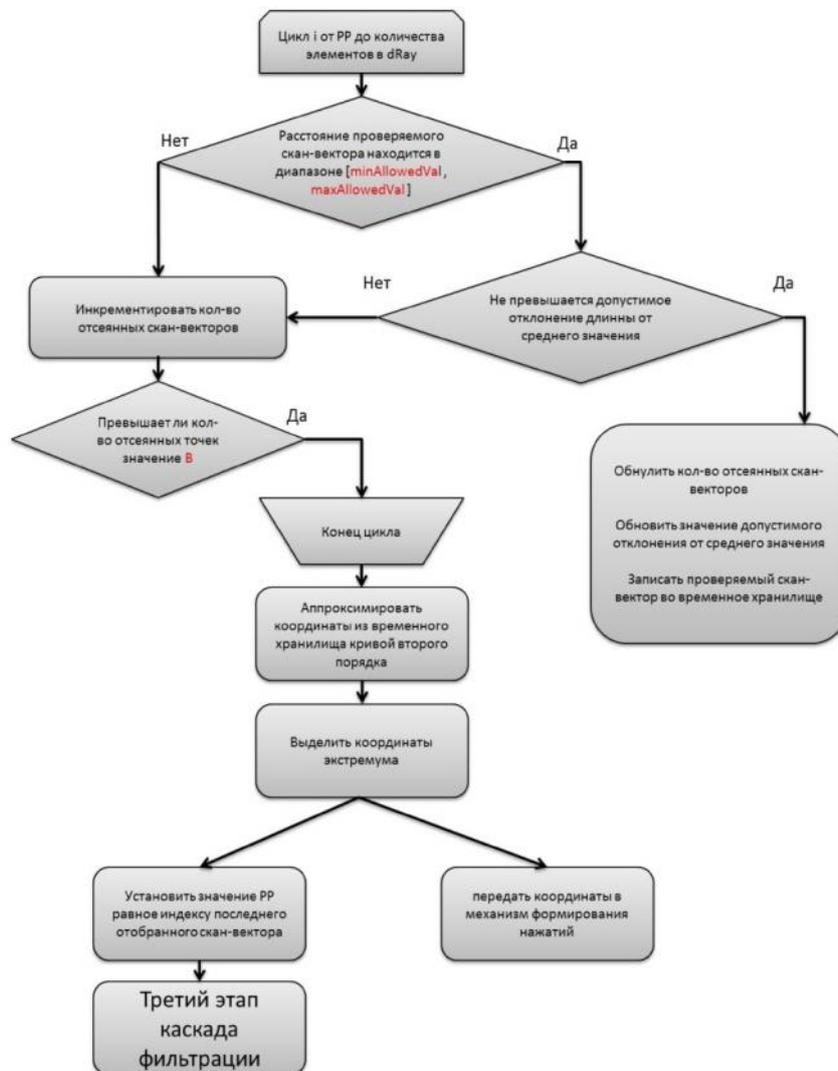   -
        ,   -   ,   ,
                                          .   -   -
                              :
-   minAllowedVal   maxAllowedVal;
-
                                        -                    .
                                                  B (B –   )   -
              .
                         -            (*PP*),   -
              .   . 6.

              .
              -                    .   -
                                      ,   -
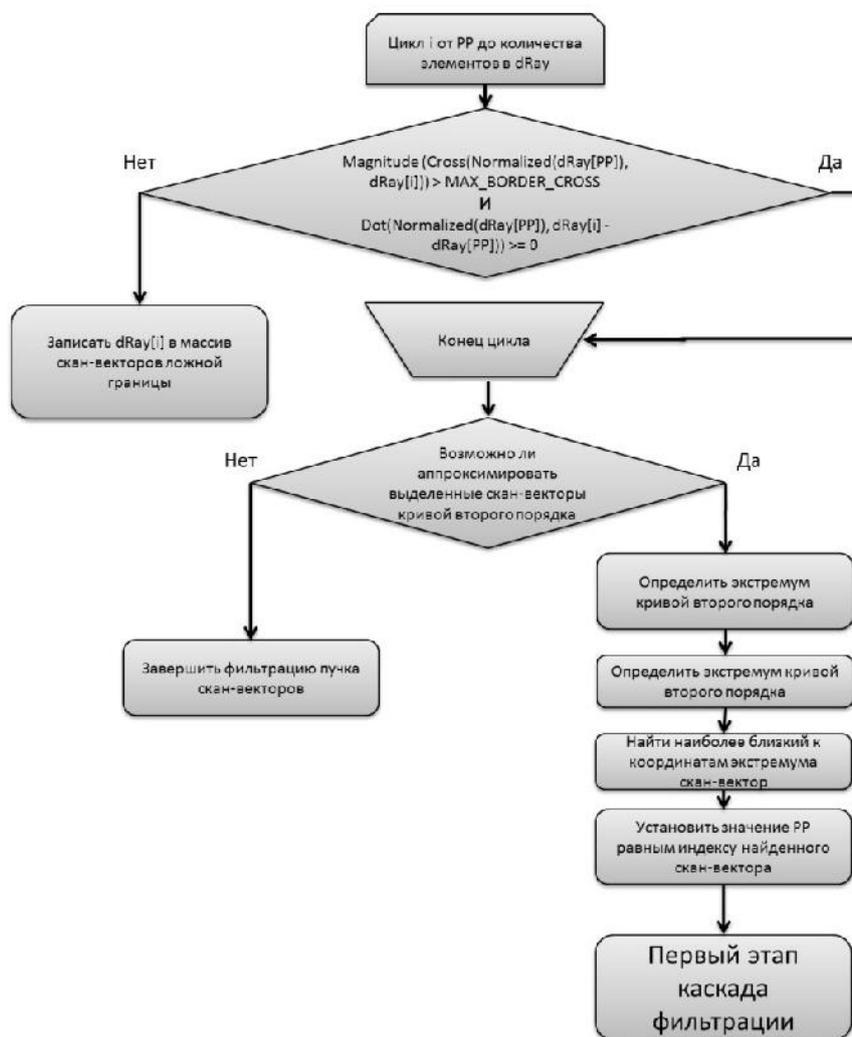   -   .

Цикл i от PP до количества элементов в dRay

Расстояние проверяемого скан-вектора находится в диапазоне [minAllowedVal, maxAllowedVal]

Нет

Да

Инкрементировать кол-во отсеянных скан-векторов

Нет

Не превышается допустимое отклонение длинны от среднего значения

Да

Превышает ли кол-во отсеянных точек значение B

Да

Обнулить кол-во отсеянных скан-векторов

Обновить значение допустимого отклонения от среднего значения

Записать проверяемый скан-вектор во временное хранилище

Конец цикла

Аппроксимировать координаты из временного хранилища кривой второго порядка

Выделить координаты экстремума

Установить значение PP равное индексу последнего отобранного скан-вектора

передать координаты в механизм формирования нажатий

Третий этап каскада фильтрации

. 6.

4.6.   .   -
                                                                              ,
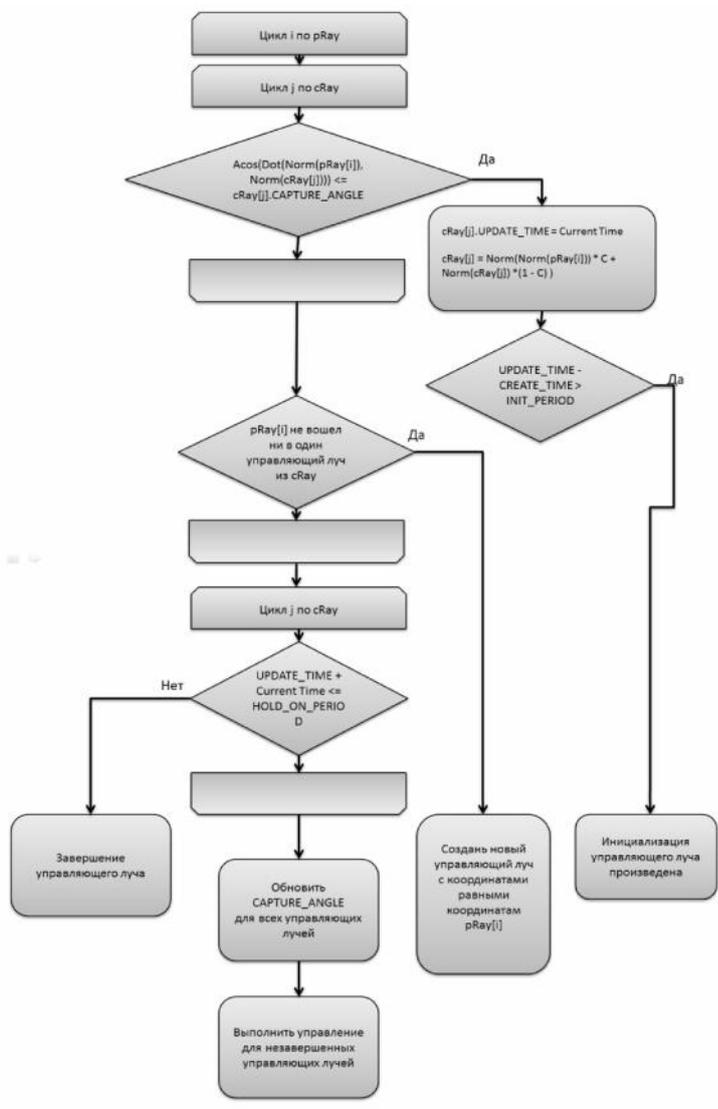   -   ,
            .   . 7.   ,   -
                              -   -   —   -
                  ,   ,   -
            -          .   -   ,   .
      ,   .

(,
 ,), ,
 - -
. , , -
- ( ),
 , *PP*
- , - -
.

Цикл i от PP до количества элементов в dRay

Magnitude (Cross(Normalized(dRay[PP]), dRay[i])) > MAX_BORDER_CROSS
И
Dot(Normalized(dRay[PP]), dRay[i] - dRay[PP])) >= 0

Нет   Да

Записать dRay[i] в массив скан-векторов ложной границы

Конец цикла

Возможно ли аппроксимировать выделенные скан-векторы кривой второго порядка

Нет   Да

Завершить фильтрацию пучка скан-векторов

Определить экстремум кривой второго порядка

Определить экстремум кривой второго порядка

Найти наиболее близкий к координатам экстремума скан-вектор

Установить значение PP равным индексу найденного скан-вектора

Первый этап каскада фильтрации

. 7.

5.

,   .

.

,   ,

(. .

-

).

,

.   ,

:

-   ,

;

-   (,   -

,   );

-   -

.

. 8.

-    -

(   )   ,   -

:

$$Acos(Dot(Norm(pRay[i]), Norm(cRay[j]))) <= cRay[j]. \ CAPTURE\_ANGLE,$$

$Acos$ –   , Dot –   , $Norm$ –   -

, $pRay$ –   -   , $cRay$ –

, $CAPTURE\_ANGLE$ –   -

,

, i j –   .

,   -

$(cRay[j]. \ UPDATE\_TIME)$ -

:

$$cRay[j] = Norm(Norm(pRay[i])) * C + Norm(cRay[j]) *(1 - C)),$$

– 0.05 0.3.

-

,   .

,   ,

$(cRay[j].$

$CREATE\_TIME)$,

$CAPTURE\_ANGLE$.

. 8.

Цикл i по pRay

Цикл j по cRay

Acos(Dot(Norm(pRay[i]), Norm(cRay[j]))) <= cRay[j].CAPTURE_ANGLE — Да

cRay[j].UPDATE_TIME = Current Time

cRay[j] = Norm(Norm(pRay[i])) * C + Norm(cRay[j]) *(1 - C))

UPDATE_TIME - CREATE_TIME > INIT_PERIOD — Да

pRay[i] не вошел ни в один управляющий луч из cRay — Да

Цикл j по cRay

UPDATE_TIME + Current Time <= HOLD_ON_PERIOD — Нет

Завершение управляющего луча

Обновить CAPTURE_ANGLE для всех управляющих лучей

Создань новый управляющий луч с координатами равными координатам pRay[i]

Инициализация управляющего луча произведена

Выполнить управление для незавершенных управляющих лучей

,   -

INIT_PERIOD.

HOLD_ON_PERIOD
(*HOLD_ON_PERIOD < INIT_PERIOD*),   .   -
.
,   -
.

-

CAPTURE_ANGLE

-

.

6.

,

, . ( ).

. -

, , .

, .

,

[4].

, -

, . , .

.

.

Blinn–Phong [5]

[6].

,

- . (

700), ,

. . ,

, — -

. -

7.

-

.

,

, .

,

, .

. -

5 .

. -

, .

. .

, -

-

.

, -

.

, .

-

. .   , . .   , . .   , . .

-

.

V.O. Petrukhin, O.S. Dzhus, S.K. Shaykhlislamov, Y.V. Necepurenko

MOBILE TECHNOLOGY OF INTERACTIVE VISUALIZATION OF THREE-DIMENSIONAL MODELS

In the article, the complex of software solutions for visualization and interactive teamwork with virtual models of medium-sized devices in full-scale mode for computer aided design problems for the purposes of scientific research & development and digital production is considered.

1.   . .   . . 2- ,   . .:   . 1987. 552 .
2.   Hussain, Zahir M., Sadik A.Z., O'Shea P. Digital signal processing: an introduction with MATLAB and applications. The First-Order Noise-Free SDPLL P. 149.
3.   Magda Y. Arduino Interfacing and SignalProcessing. (Kindle Edition), 2012.
4.   Hussain, Zahir M., Sadik A.Z. Amin Unity 5.x Shaders and Effects Cookbook. *Creating a transparent material.* P. 40
5.   Hussain, Zahir M., Sadik A.Z. Amin Unity 5.x Shaders and Effects Cookbook. *Creating a BlinnPhong Specular type.* P. 68.
6.   Hussain, Zahir M., Sadik A.Z. Amin Unity 5.x Shaders and Effects Cookbook. *Screen Effects with Unity Render Textures.* P. 151.
7.   Richard Blum, C# Network Programming. *Analyzing Network Packets.* P. 51.
8.   Richard Blum, C# Network Programming. *Programming with TCP and UDP 69.*

11.06.2018

Об авторах:

,

. .   ,

-                                    (   ),

-mail: vapetr@gmail.com

,

-                                    (   ),

-mail: dzhus@phystech.edu

,

-                                    (   ),

-mail: sha-sabir@yandex.ru

,

-                                    (   ).

-mail: evgenech92@gmail.com