

Задачи динамического поиска на поверхностях в трёхмерном пространстве

Павел Воронин

Факультет Вычислительной Математики и Кибернетики
Московский Государственный Университет им. Ломоносова
pavel.voronin@gmail.com

Аннотация

В статье рассматриваются задачи поиска подвижных объектов на поверхностях в трёхмерном пространстве. Предложен численный метод расчёта информационных множеств для таких задач. Метод основан на представлении множеств при помощи полей расстояний. Выполнение геодезических операций над множествами сводится к нахождению вязкостного решения уравнения Гамильтона-Якоби. При помощи разработанного аппарата решается задача нахождения достаточных поисковых ресурсов, задача построения поисковой траектории и задача нахождения достаточного числа ищущих.

Keywords: динамический поиск, поля расстояний, геодезические расстояния.

1. ВВЕДЕНИЕ

Данная работа посвящена геометрическому подходу к решению задач поиска активно уклоняющихся объектов. Базовой идеей методов этого типа является построение переменных во времени информационных множеств, описывающих процесс накопления ищущими и уклоняющимися объектами сведений о положении друг друга. Обширный обзор работ по данной теме дан в работе [1].

Как показала практика, аналитическое построение информационных множеств для нетривиальной структуры области поиска или большого количества ищущих объектов сопряжено со значительными трудностями. В связи с этим в диссертации [2] было предложено перейти к графическим методам решения поисковых задач: рассчитывать информационные множества на компьютере при помощи специальных численных методов с сопутствующей динамической визуализацией.

Для случая поиска на плоскости описано два подхода к построению численных методов расчёта информационных множеств: 1) в работе [2] был предложен метод, основанный на представлении поискового множества набором

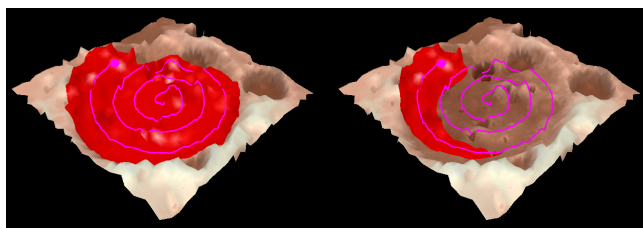


Figure 1: Поиск на ландшафте. Красным выделено множество, гарантированно свободное от искомых объектов (при двух разных соотношениях скоростей ищущего и искомого объектов).

отрезков, аппроксимирующих его границу; 2) в работе [3] используется неявное задание множеств посредством полей расстояний [4].

В данной статье описывается подход, позволяющий расширить применимость метода, основанного на использовании полей расстояний, на случай поиска на поверхностях в трёхмерном пространстве (рис. 1). Для вычисления возникающих при этом геодезических расстояний предлагается воспользоваться аппаратом численного решения уравнений Гамильтона-Якоби [5].

На основе описанного подхода были разработаны подходы к решению трёх классов задач: 1) задачи нахождения поисковых траекторий (перебором из заданного класса); 2) задачи отыскания достаточных поисковых ресурсов (скоростей ищущих, дальности видимости); 3) задачи нахождения достаточного количества ищущих.

2. ОСНОВНЫЕ ПОНЯТИЯ. ФОРМУЛИРОВКА ЗАДАЧИ

Рассмотрим задачу поиска объектов на заданном множестве. Воспользовавшись классификацией из статьи [1], введём основные понятия.

Область поиска называется некоторое множество, на котором допускается нахождение участвующих в поиске объектов.

Объекты поиска (искомые объекты) могут быть статичными или подвижными. Для простоты их можно считать точечными.

Ищущие объекты — это подвижные точки, в задачу которых входит отыскание такого обхода области поиска, при котором будут обнаружены все объекты поиска или же будет доказано, что ни одного искомого объекта в заданной области не было.

В качестве *условия обнаружения* будем рассматривать приближение одного из ищущих к искомому на заданное расстояние.

В зависимости от задачи, поведение искомого объектов может быть разным. Мы будем рассматривать случай, когда искомые объекты активно уклоняются от ищущих, местоположение, скорость и стратегия движения которых в каждый момент времени им известны, — в то время как ищущие знают только максимальную скорость уклоняющихся. Эту же ситуацию с точки зрения ищущих можно интерпретировать иначе: ничего, кроме максимальной скорости противников, ни одной из сторон неизвестно — и, в дополнение к этому, ищущим неизвестно количество искомого объектов. Выбираемая стратегия должна гарантировать нахождение сколь угодно большого конечного числа объектов.

Геометрический подход к решению поисковых задач основан на использовании вспомогательных переменных во

времени *информационных множеств*, соответствующих сведениям, накапливаемым в процессе поиска ищущими и уклоняющимися объектами друг о друге. Наиболее значимым типом информационных множеств является *остаточная область* — множество точек области поиска, в которых уклоняющихся объектов не может быть (иначе они были бы обнаружены ищущими ранее). Целью ищущих является увеличение остаточной области вплоть до момента, когда она охватит всю область поиска.

Информационные множества показали себя как крайне эффективный инструмент при решении многих задач динамического поиска. Вместе с тем, попытки применить их для случая нескольких ищущих объектов или невыпуклых поисковых областей сталкиваются с серьёзными трудностями. Дело в том, что структура информационных множеств в этих случаях значительно усложняется [1] и аналитические методы их построения становятся неэффективными. Поэтому для дальнейшего развития геометрического подхода к исследованию поисковых задач потребовалась разработка численных методов нахождения информационных множеств.

3. ОБЩАЯ СХЕМА ЧИСЛЕННЫХ МЕТОДОВ

Рассмотрим общую схему построения численных методов для расчёта остаточной области [2]. Для простоты ограничимся случаем области поиска без границы, одного ищущего и одного уклоняющегося объекта, движущихся с постоянными скалярными скоростями.

Введём следующие обозначения. Пусть скалярная скорость уклоняющегося объекта равна U , а скорость ищущего объекта — V (будем считать, что $V > U$, то есть ищущий имеет преимущество по скорости). Пусть область обнаружения ищущего — круг радиуса R , а траектория его перемещения $P(t)$. Нам нужно рассчитать изменение переменной остаточной области $A(t)$ на отрезке времени от $t = 0$ до $t = T$.

Дискретизируем задачу. Приближим заданную траекторию ломаной, проходящей через точки P_0, P_1, \dots, P_n — положения ищущего в моменты времени $t_0 = 0, t_1, \dots, t_n = T$ ($t_i = i\Delta t, \Delta t = T/n$). Будем последовательно искать остаточные области $A_0, A_1, \dots, A_n = A$, где $A_i = A(t_i)$. Как показано в работе [3], общая формула описания фигур A_k выглядит так:

$$\begin{cases} A_0 = B(P_0; R), \\ A_{k+1}^{old} = Offset(A_k, -\Delta U), \\ A_{k+1}^{new} = A(P_k, P_{k+1}, r, R), \\ A_{k+1} = A_{k+1}^{old} \cup A_{k+1}^{new}. \end{cases} \quad (1)$$

Здесь $B(P_0; R)$ — это круг радиуса R с центром в точке P_0 ; $Offset(A, -r)$ — сужение множества A на полосу шириной r (r -расширение с отрицательным r); $A(P_k, P_{k+1}, r, R)$ — т.н. фигура A -типа: $A(P_k, P_{k+1}, r, R) = \bigcup_{t \in [t_k, t_{k+1}]} Offset(B(P(t), R), -U(t - t_k))$. На практике вместо точного значения $A(P_k, P_{k+1}, r, R)$ может использоваться некоторое его приближение, например $A(P_k, P_{k+1}, r, R) \approx B(P_{k+1}, R)$ или $A(P_k, P_{k+1}, r, R) \approx \bigcup_{i=0}^m B^U(P(t^i)); t \in [t_k, t_{k+1}], \forall i$, где $B^U = Offset(B(P(t^i), R), -U(t^i - t_k))$.

Обобщения вышеприведённых формул на случай нескольких ищущих и областей поиска с границей описаны в статье [3] и опираются на те же самые базовые операции.

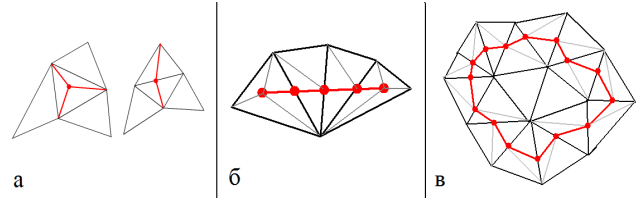


Figure 2: Подразбиение сетки: а) по точке; б) по отрезку; в) по линии уровня.

4. МЕТОДЫ, ОСНОВАННЫЕ НА ИСПОЛЬЗОВАНИИ ПОЛЕЙ РАССТОЯНИЙ

Итак, для широкого класса поисковых задач возможно приближённое рекуррентное вычисление информационных множеств. Заметим, что для реализации численного метода на их основе нам необходимо в каждом случае выбирать такое программное представление множеств, которое позволяло бы эффективно проводить над ними следующие четыре базовые операции: объединение, пересечение, разность и r -расширение.

В работе [3] в качестве такого представления предложено неявное задание множеств, известное как поле расстояний (Distance Field, DF [4]): вместо заданного замкнутого геометрического объекта будем рассматривать порождаемое им скалярное поле, значение которого в произвольной точке равно расстоянию от этой точки до границы объекта, взятому со знаком плюс, если точка лежит вне объекта, и со знаком минус в противном случае.

Покажем, как при помощи полей расстояний проводятся вычисления. Возьмём последнее уравнение системы 1 и перепишем его, выразив все действия через операции над полями расстояний. Получим следующую последовательность действий (DF -алгоритм) [3]:

- 1) $DF_{old} = DF_{A_k} + \Delta U$,
- 2) $DF_{upd} = DF_{A(P_k, P_{k+1}, r, R)}$,
- 3) $DF_{new} = \min(DF_{old}, DF_{upd})$,
- 4) $DF_{A_{k+1}} = DF_{ISO_0 DF_{new}}$.

5. ИНФОРМАЦИОННЫЕ МНОЖЕСТВА НА ПОВЕРХНОСТЯХ

Поисковым задачам на поверхностях в трёхмерном пространстве посвящено значительное количество работ, однако большинство из них посвящено достаточно простым аналитически заданным поверхностям. Например, в работе [6] описаны поисковые траектории шаре, в работе [7] — на цилиндре и торе, а в диссертации [8] — на поверхностях вращения. Целью настоящей работы является разработка метода, который бы позволил решать поисковые задачи на поверхностях достаточно общего вида.

В качестве области обнаружения в задачах на поверхностях обычно рассматривают геодезический круг с центром в точке расположения ищущего. Под r -расширением множества на поверхностях понимается его геодезическое r -расширение, то есть сумма (или разность) Минковского этого множества и всех геодезических кругов радиуса r с центрами на границе множества.

Таким образом, для вычисления информационных множеств по приведённому выше алгоритму нам понадобится аппарат быстрого вычисления геодезических расстояний

ний и геодезических r -расширений.

5.1 Геодезические расстояния и методы их вычисления

Пусть на некоторой поверхности в трёхмерном пространстве задана область Ω и введена метрика $g : x \in \Omega \mapsto H(x) \in R^{2 \times 2}$, $H(x) > 0$. Пусть в области Ω задана некоторая регулярная кривая $\gamma(t)$, $t \in [0, 1]$. Длина этой кривой вычисляется по следующей формуле [9]: $L(\gamma) = L_g(\gamma) = \int_0^1 \sqrt{(\gamma'(t))^T H(\gamma(t)) \gamma'(t)} dt$.

В области Ω расстоянием $D(A, B)$ между точками A и B называется наименьшая из длин соединяющих их кривых, целиком лежащих в данной области.

Функционалом длины для множества точек $S = \{x_k\}_{k=1}^N \subset \Omega$ называется функция $U_S(x) = \min_k D(x, x_k)$, $x \in \Omega$. Имеет место следующая теорема [10, 11].

Теорема 1. Для любого множества $S = \{x_k\}_{k=1}^N \subset \Omega$ функционал длины $U_S(x)$ есть единственное вязкостное решение уравнения Гамильтона-Якоби

$$\|\nabla U_S\|_{H^{-1}} = 1 \text{ при условии } U_S(x_k) = 0, \forall k, \quad (2)$$

где $\|v\|_A = \sqrt{v^T A v}$.

Для вычисления функционала длины разработан целый ряд эффективных численных методов. Большинство из них работают с сеточным приближением задачи: по заданной сетке (и значениям метрики в её узлах) находят для любого подмножества узлов S приближённое значение функционала длины U_S во всех узлах этой сетки. Наиболее распространённым подходом является численное решение уравнения (2) посредством одной из модификаций метода быстрого движения (Fast Marching, FM [5]). При использовании подходящих эффективных структур данных его сложность составляет $O(n \log(n))$, где n — число вершин сетки. На практике алгоритм ведёт себя почти линейно. В статье [12] приводится модификация FM-алгоритма, в которой при помощи квантования приоритета удаётся добиться снижения теоретической сложности до оптимальной $O(n)$.

При реализации описанных в данной работе алгоритмов мы опирались на вариант FM-алгоритма, входящий в состав библиотеки *Geowave*, являющейся частью *Toolbox Fast Marching*¹. При помощи этого же инструментария мы генерировали равномерные сетки для всех исследуемых поверхностей, следуя алгоритму ретриангуляции, изложенному в статье [13].

5.2 Геодезический поиск

Опишем реализацию DF-алгоритма для построения остаточной области на поверхностях в трёхмерном пространстве.

1. Первый и третий шаги DF-алгоритма выполняются за линейное от количества узлов сетки время.
2. В соответствии с первой строкой системы 1, для нахождения фигуры A_0 необходимо вычислить поле расстояний для геодезической окружности с заданным центром и радиусом: $DF_{B(C;R)}(x) = DF_C(x) -$

R . Оно выражается через поле расстояний для центра окружности, а поле расстояний для точки есть её функционал длины. Значит, его можно найти при помощи FM-алгоритма. Заметим, однако, что в классическом варианте при помощи алгоритма быстрого движения можно найти функционалы длины только для точек, являющихся узлами сетки. Если же точка C попадает внутрь треугольника сетки или на его ребро, понадобится предварительно сделать подразбиение сетки (рис. 2а).

3. На втором шаге DF-алгоритма необходимо вычислить поле расстояний для фигуры $A(P_k, P_{k+1}, r, R)$. Сначала рассмотрим случай $r = R$. Тогда $A(P_k, P_{k+1}, R, R) = \bigcup_{t \in [t_k, t_{k+1}]} B(P(t), R)$, то есть $A(P_k, P_{k+1}, R, R)$ — геодезическое R -расширение криволинейного отрезка $[P_k, P_{k+1}]$. Следовательно, $A(P_k, P_{k+1}, R, R) = ISO_R DF_{[P_k, P_{k+1}]}$. Подразбив по $[P_k, P_{k+1}]$ сетку (рис. 2б), получим набор новых точек $S^* = \{s_i^*\}_{i=1}^m$. Имеет место следующее приближение: $A(P_k, P_{k+1}, R, R) \approx ISO_0(\min_i DF_{B(s_i, R)})$. Возвращаясь к общему случаю, по аналогии имеем

$$A(P_k, P_{k+1}, r, R) \approx ISO_0(\min_i DF_{B(s_i, r_i)}),$$

$$r_i = r + (R - r) \frac{L([s_i, P_k])}{L([P_{k+1}, P_k])}.$$

Таким образом, мы выразили границу фигуры $A(P_k, P_{k+1}, r, R)$ через поверхность уровня некоторого поля.

4. На четвёртом шаге DF-алгоритма и в конце п. 3 требуется вычислять поле расстояний для области K , граница которой задана нулевой изолинией некоторого поля, $K = ISO_0 D$, а значит, состоит из отрезков, вершины которых лежат на рёбрах сетки. Как в п.п. 2-3, подразобьём сетку по этим отрезкам (рис. 2в) и получим набор новых узлов S^* . Легко видеть, что $DF_K = \text{sgn}(D) \cdot U_{S^*}$.
5. Заметим также, что FM-алгоритм позволяет вычислять функционал длины не на всей сетке, а лишь на тех её узлах, где его значение не превышает некоторой заданной величины. Это позволяет значительно ускорить работу алгоритма: будем вычислять значение поля расстояний лишь в узлах сетки, удалённых от нулевой изолинии менее, чем на $V \cdot \Delta t + e_{max}$ (e_{max} — максимальная длина ребра сетки).

Описанный алгоритм был реализован на языке C++ и позволяет вычислять и визуализировать (с использованием библиотеки OpenGL) остаточные области для траекторий на поверхностях в реальном времени. Например, анимация, кадр из которой приведён на рис. 1, рассчитывалась и отрисовывалась со скоростью 40-50 кадров в секунду на обычном домашнем компьютере (Pentium 4 3 GHz, 1.5 Gb DDR 400 MHz, NVidia GeForce 6800) при размерах сетки в несколько тысяч треугольников.

6. РЕШЕНИЕ ПОИСКОВЫХ ЗАДАЧ

Разработанный аппарат позволяет в интерактивном режиме решать разнообразные поисковые задачи, сводящиеся к вычислению информационного множества для некоторого набора траекторий и/или ресурсов.

¹<http://www.ceremade.dauphine.fr/~peyre/matlab/>

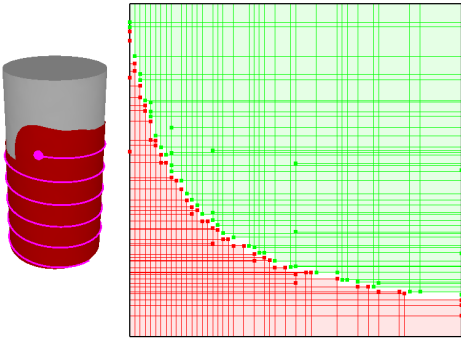


Figure 3: Поиск на цилиндре. На графике зелёным отмечены области фазовой плоскости, которым соответствуют ресурсы, гарантированно достаточные для успешного поиска, красным — гарантированно недостаточные.

Простейшей из таких задач является задача определения, гарантирует ли заданная траектория движения ищущего объекта обнаружение активно уклоняющихся объектов при известных скоростях всех участников и известном радиусе обнаружения. Чтобы ответить на этот вопрос, достаточно построить остаточную область и проверить, совпадает ли она по окончании движения со всей областью поиска. Так же решается и задача для нескольких ищущих.

Несколько более трудными являются задача нахождения достаточных ресурсов, задача построения поисковой траектории и задача нахождения достаточного числа ищущих объектов: для их решения необходим перебор наборов ресурсов, траекторий и ищущих соответственно. Опишем методы решения каждой из этих задач, рассматривая для простоты изложения поверхности тел простой структуры (те же самые подходы применимы и для поверхностей общего вида, однако соответствующие траектории могут быть устроены гораздо сложнее).

6.1 Нахождение достаточных ресурсов

Рассмотрим задачу гарантированного поиска активно уклоняющихся объектов в некоторой заданной ограниченной области. Пусть траектории ищущих фиксированы. Ищущие обладают определёнными поисковыми ресурсами. Для любого набора ресурсов, пользуясь описанным выше алгоритмом, мы можем построить остаточную область и определить, совпадает ли она с областью поиска или нет, а значит, являются ли ли данные ресурсы достаточными для нахождения уклоняющихся объектов при движении по заданной ломаной. Решая эту задачу для целого ряда наборов, мы можем охарактеризовать ресурсы, достаточные для гарантированного нахождения.

В качестве примера рассмотрим фиксированную винтовую линию на цилиндре. В качестве ресурсов будем рассматривать радиус круга обнаружения R и отношение скорости ищущего объекта к скорости уклоняющихся U . Пусть ресурсы могут изменяться в пределах $R \in [R_{min}, R_{max}]$, $U \in [U_{min}, U_{max}]$.

Заметим, что наша задача обладает свойством монотонности: если какая-то кривая является решением при ресурсах (R^*, U^*) , то она является решением и при любых ресурсах (R^{**}, U^{**}) таких, что $R^{**} \geq R^*$, $U^{**} \geq U^*$

(это следует из вложенности соответствующих остаточных областей). В терминах фазовой плоскости $R - U$ это означает, что достаточность ресурсов (R^*, U^*) влечёт за собой достаточность из квадрата $[R^*, R_{max}] \times [U^*, U_{max}]$. Верно и обратное: из недостаточности ресурсов (R^*, U^*) следует недостаточность ресурсов из квадрата $[R_{min}, R^*] \times [U_{min}, U^*]$. Эти свойства можно выразить и по-другому: достаточные ресурсы на фазовой плоскости лежат в области, ограниченной снизу графиком неубывающей функции, $\{(R, U) \mid U \geq F(R)\}$. Приближённое отыскание этой кривой, в свою очередь, можно свести к набору одномерных задач.

Будем искать приближённые значения $U_i \approx F(R_i)$, $1 \leq i \leq N$, $|R_i - R_{i+1}| < \varepsilon_R$, $|U_i - F(R_i)| < \varepsilon_U$. Припишем каждому значению U_i интервал возможных значений $[U_i^1, U_i^2]$. В начальный момент все интервалы равны, $U_i^1 = U_{min}$, $U_i^2 = U_{max}$. Когда для какого-нибудь значения U_i интервал уточняется, то по свойству монотонности уточняются и интервалы всех остальных значений: $[U_i^1, U_i^2] \rightarrow [V^1, V^2] \Rightarrow 1) [U_j^1, U_j^2] \rightarrow [\max(U_j^1, V^1), U_j^2]$, $\forall j < i$; 2) $[U_j^1, U_j^2] \rightarrow [U_j^1, \min(U_j^2, V^2)]$, $\forall j > i$.

Для нахождения U_i , отстоящего от $F(R_i)$ не более чем на ε_U , нам достаточно заключить его в интервал длины не более $2\varepsilon_U$, $U_i^2 - U_i^1 \leq 2\varepsilon_U$. Будем делать это методом дихотомии. На каждой итерации будем брать ресурсы (R_i, U^*) , $U^* = \frac{U_i^1 + U_i^2}{2}$. Если таких ресурсов достаточно для решения задачи, то $[U_i^1, U_i^2] \rightarrow [U_i^1, U^*]$, иначе $[U_i^1, U_i^2] \rightarrow [U^*, U_i^2]$. Если получившийся интервал имеет длину менее $2\varepsilon_U$, то заканчиваем вычисления. Используя полученный интервал уточняем интервалы для значений $U_j, j \neq i$, а за значение U_i берём середину полученного интервала.

Поскольку уточнение интервала значений одного из U_i влияет на все остальные, то имеет значение порядок их вычисления. Разумным видится следующий порядок: $U_1, U_N, U_{\frac{N}{2}}, U_{\frac{N}{4}}, U_{\frac{3N}{4}}, U_{\frac{N}{8}}, U_{\frac{3N}{8}}, U_{\frac{5N}{8}}, U_{\frac{7N}{8}}$ и т. д. Практика показывает, что в большинстве случаев уточнение интервалов значений при описанном обходе существенно уменьшает количество итераций. Например, для $U, R \in [0, 1]$, $\varepsilon_U = \varepsilon_R = \frac{1}{100}$, $F(R) = 1 - R$ алгоритму требуется 322 шага, тогда как при независимом вычислении U_i потребовалось бы порядка 700 шагов.

Работа алгоритма проиллюстрирована на рис. 3.

6.2 Построение поисковой траектории

Ещё одной задачей, легко сводящейся к нахождению остаточных областей, является выбор из некоторого заданного класса кратчайшей траектории, обеспечивающей нахождение активно уклоняющихся объектов. В качестве примера рассмотрим задачу построения поисковой траектории для поверхности тела, обладающего осевой звёздностью (при заданных скоростях объектов и радиусе обнаружения). Без ограничения общности будем считать, что поверхность задана уравнениями:

$$\{x = R(h, \phi) \cos(\phi); y = h; z = R(h, \phi) \sin(\phi)\},$$

где $h \in [0, 1]$, $\phi \in [0, 2\pi]$, $R: [0, 1] \times [0, 2\pi] \rightarrow [R_{min}, R_{max}]$.

Будем искать траекторию в классе обобщённых винтовых линий на этой поверхности:

$$\{x = R(h, \Phi(h)) \cos(\Phi(h)); y = h; z = R(h, \Phi(h)) \sin(\Phi(h))\},$$

где $h \in [0, 1]$, а $\Phi(h): [0, 1] \rightarrow [0, +\infty]$ — неубывающая

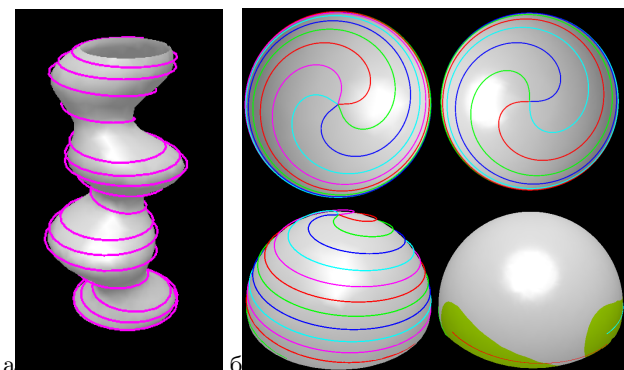


Figure 4: а) Поисковая траектория на звёздчатом многограннике; б) Поиск несколькими ищущими.

функция.

Разобьём отрезок $[0, 1]$ на N_h равных частей: $h_0 = 0, h_1 = \Delta h = \frac{1}{N_h}, h_2 = 2\Delta h, \dots, h_{N_h} = 1$. Последовательно на каждом отрезке $[h_i, h_{i+1}]$ будем искать $\Phi(h)$ в виде линейной функции: $\Phi(h) = \Phi(h_i) + \alpha_i(h - h_i)$, где $\alpha_i \in [\alpha_{min}, \alpha_{max}]$. Параметр α_i будем искать методом дихотомии. Приближение $\tilde{\alpha}_i \approx \alpha_i$ будем считать оценкой сверху, если для кривой, заданной $\alpha_0, \alpha_1, \dots, \alpha_{i-1}, \tilde{\alpha}_i$ при $h \in [0, h_{i+1}]$, остаточная область покрывает часть поверхности, заключённую между плоскостями $y = 0$ и $y = h_{i+1}$, — и оценкой снизу в противном случае.

Такой жадный алгоритм с одной стороны гарантирует быстрое нахождение поисковой траектории, а с другой — обеспечивает адаптацию кривой к конкретному виду поверхности: витки винтовой линии будут плотнее упакованы на участках, лежащих близко к оси, и реже — на лежащих далеко от неё.

Результат работы алгоритма для некоторого тела, обладающего осевой звёздностью, приведён на рис. 4а).

6.3 Нахождение достаточного числа ищущих

Если для некоторой области и одного ищущего с заданными ресурсами не удаётся построить поисковую траекторию, которая бы обеспечивала гарантированное нахождение искомого объекта, то встаёт вопрос о числе ищущих, при котором это возможно (и соответствующих кратчайших траекториях). Если для любого числа ищущих мы можем указать достаточно представительный класс поисковых траекторий и порядок их перебора, то такую задачу можно решать тем же методом, что и описанные выше.

В качестве примера рассмотрим полусферу достаточно большого радиуса. Для любого числа ищущих n будем рассматривать поисковые траектории, представляющие из себя спирали, начинающиеся в вершинах вписанного в основание полусферы правильного n -угольника, сходящиеся на полюсе и имеющие одинаковые углы наклона $\alpha \in [\varepsilon_\alpha, \frac{\pi}{2}]$ (см. рис. 4б). Для нахождения минимального числа ищущих достаточно рассматривать траектории с углом наклона ε_α . Кратчайшие траектории из соответствующего класса можно найти, перебирая углы наклона методом дихотомии (см. п.п. 6.1, 6.2).

Работа алгоритма проиллюстрирована на рис. 4б).

7. REFERENCES

- [1] Чхартишвили А. Г. and Шикин Е. В., “Динамический поиск объектов. Геометрический взгляд на проблему,” *Фундаментальная и прикладная математика*, vol. 1, no. 4, pp. 827–862, 1995.
- [2] Березин С. Б., *Графический подход к исследованию и решению задач динамического поиска*, Ph.D. thesis, МГУ, 2000.
- [3] Воронин П., “О новом подходе к расчёту и визуализации информационных множеств в задачах динамического поиска,” in *Труды 18-й международной конференции ГрафиКон*. Московский Государственный Университет, 2008.
- [4] В. А. Payne and A. W. Toga, “Distance field manipulation of surface models,” *Computer Graphics and Applications*, vol. 12, no. 1, pp. 65–71, 1992.
- [5] J. A. Sethian, *Level Sets Methods and Fast Marching Methods. 2nd edition*, Cambridge University Press, 1999.
- [6] Залгаллер В. А., “Кратчайшие линии осмотра сферы,” *dep. в ВИНТИ 24.09.92, № 2844-B92*.
- [7] Губайдуллин С. М. and Шикин Е. В., “Следящие области на цилиндре и на торе,” *Вестник московского университета. Сер. 15. Вычислительная математика и кибернетика*, vol. 0, no. 2, pp. 46–50, 1992.
- [8] Чхартишвили А. Г., *Метод следящих областей в задачах динамического поиска*, Ph.D. thesis, МГУ, 1994.
- [9] Карган Э. Ж., *Риманова геометрия в ортогональном репере*, МГУ, 1960.
- [10] Курант Р., *Уравнения с частными производными*, Мир, 1964.
- [11] M. G. Crandall, H. Ishii, and P. L. Lions, “User’s guide to viscosity solutions of second order partial differential equations,” *Bulletin of the American Mathematical Society*, vol. 27, no. 1, pp. 1–67, 1992.
- [12] L. Yatziv, A. Bartesaghi, and G. Sapiro, “O(n) implementation of the fast marching algorithm,” *Journal of Computational Physics*, vol. 212, no. 2, pp. 393–399, 2006.
- [13] G. Peyré and L. D. Cohen, “Geodesic remeshing using front propagation,” *International Journal on Computer Vision*, vol. 69, no. 1, pp. 145–156, 2006.

ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ

Расширенная версия данной статьи и дополнительные цветные анимированные иллюстрации, демонстрирующие работу описанных алгоритмов, доступны на сайте <http://www.primaler.ru/cg/>. Там же доступна для скачивания реализация описанных алгоритмов на C++ (платформено-независимая) и MATLAB. Отдельный раздел сайта посвящён ссылкам на литературу по теме.