

ОПТИМИЗАЦИЯ РАЗМЕЩЕНИЯ СРЕДСТВ НАБЛЮДЕНИЯ В ТРЁХМЕРНОЙ СЦЕНЕ С ЦЕЛЮ МИНИМИЗАЦИИ «СЛЕПЫХ ЗОН»

В.В. Печенкин¹, М.С. Королёв¹

¹ Саратовский государственный технический университет имени Гагарина Ю.А., Саратов, Россия

Аннотация

Рассматривается применение средств наблюдения различных типов и их конфигурации в разработанном программном комплексе, предназначенном для минимизации «слепых зон» в сложной трёхмерной сцене. Описывается архитектура программного комплекса, принцип работы средств наблюдения, алгоритм определения слепых зон. Формальная постановка задачи минимизации «слепых зон» сведена к решению оптимизационной задачи для специально определённого графа наблюдаемости.

Ключевые слова: средства наблюдения, аудиальные сенсоры, визуальные сенсоры, оптимизация расположения, наблюдаемость.

Цитирование: Печенкин, В.В. Оптимизация размещения средств наблюдения в трёхмерной сцене с целью минимизации «слепых зон» / В.В. Печенкин, М.С. Королёв // Компьютерная оптика. – 2017. – Т. 41, № 2. – С. 245-253. – DOI: 10.18287/2412-6179-2017-41-2-245-253.

Введение

В статье рассматривается задача оптимизации размещения средств наблюдения различного типа в трёхмерной сцене. Под трёхмерной сценой понимается виртуальная модель реальной площадки с размещёнными на ней объектами, за которыми необходимо установить наблюдение. Сама сцена имеет препятствия для средств наблюдения либо в виде геометрических трёхмерных объектов, либо в виде поверхностей. При выборе позиций для размещения средств наблюдения (будем далее называть их камерами или сенсорами) на сцене образуются так называемые «слепые зоны», которые должны быть минимизированы. Решение задачи в этой постановке основывается на критерии оптимальности с точки зрения количественных характеристик слепых зон на сцене. В данной работе не рассматривается задача подбора оборудования для наблюдения по его характеристикам, однако эта проблема является актуальной и её решение требует отдельного внимания [1].

Кратко опишем основные существующие в настоящее время подходы. Одной из частых проблем при решении подобных задач, а именно оптимизации размещения сенсоров, является необходимость регулярного выполнения перекалибровки устройств с целью определения их параметров. С подобными вопросами помогают справиться многоэтапные алгоритмы самокалибровки [2], позволяющие вычислять матрицу камеры, а также её основные характеристики на основе реконструкции поверхности. Применение фильтрующих алгоритмов в совокупности с анализом результатов предыдущих калибровок позволяет «экстраполировать» характеристики камеры, если необходимо добиться быстрого расчёта в определённый момент времени [3].

Динамический характер объектов наблюдения требует отдельного подхода [4]. Позиции камер могут вычисляться как на плоскости, так и в трёхмерном пространстве на основе данных о перемещениях объектов на текущий момент времени. При этом направление движения обычно считается известным заранее. Развитие такого решения позволяет определять ка-

либровочные характеристики для фиксированных камер на основе данных о перемещениях в зонах высокой активности [5].

В случаях, когда расположение объектов-целей на сцене заранее неизвестно и их необходимо обнаружить путём обследования всей сцены, средства наблюдения должны быть размещены так, чтобы максимально покрыть всю область наблюдения, минимизировав количество «слепых зон» – областей, которые не наблюдаются ни одним сенсором. Сцена при этом может иметь сложную геометрическую структуру, что должно учитываться при вычислениях. Одним из эффективных способов выполнения данной задачи является решение оптимизационной задачи эффективного размещения средств наблюдения, которое основывается на приближённых методах поиска оптимальных решений [6].

При отсутствии информации об объектах наблюдения необходимо получать её на основе реальных данных, которые являются результатом сканирования сцены сенсорами. Первой задачей в таком случае будет определение расстояния между целью и средством наблюдения, и используемый алгоритм определения оптимального положения камер должен учитывать комплексное расположение средств наблюдения в сцене с целью минимизации ошибки определения дистанций [7]. Когда система, управляющая средствами наблюдения, не имеет предварительной информации о размещении целей и препятствий в сцене, трёхмерная реконструкция сцены является необходимым решением для последующей оптимизации положения камер [8, 9].

Когда наблюдение происходит за сценами, имеющими большую площадь, требуется большое количество камер (несколько десятков или сотен). Сама размерность задачи не позволяет руководствоваться традиционными способами решения. В таких случаях удобно использовать трёхмерное моделирование сцены, на которой предстоит решать задачу, и вычисления производятся на этой модели. В работе [10] предложено именно такое решение, которое для трёхмер-

ной модели здания формирует оптимальное положение набора из 400 камер наблюдения в целях максимизации площади покрытия зоны наблюдения.

Возможны ситуации, когда необходимо произвести эту же работу на открытой местности, имеющей не менее сложные геометрические параметры. В работе [11] рассматривается способ определения оптимальной конфигурации системы видеомониторинга леса для обнаружения и определения местоположения лесных пожаров. Такие системы применяются на практике сравнительно недавно, их актуальность возрастает, поскольку проблема является одной из наиболее серьёзных и не решённых на сегодняшний день. В этом же русле решается задача в работе [12], дающей описание алгоритма нахождения позиций для размещения средств наблюдения на холмистом ландшафте, что позволяет минимизировать их количество.

Одним из подвидов задач оптимизации размещения подобных сенсоров является задача оптимального размещения на границе контролируемой области. Благодаря использованию комбинаторных методов данный подход позволяет использовать вероятностные модели локализации проникновения на охраняемую территорию [13].

При рассмотрении слепых зон, как правило, рассматриваются соответствующие задачи определения слепых зон транспортных средств, вероятности нахождения объектов в данных зонах в реальном времени на основе изображения со средств видеофиксации [14, 15].

В данной работе предлагается описание алгоритма вычисления слепых зон для трёхмерной сцены и заданной конфигурации расположения камер наблюдения, описывается структура программного пакета, в котором реализуется этот алгоритм, и результатов проведённого вычислительного эксперимента. Некоторые предварительные идеи и определения, используемые нами, представлены в работах [16, 17].

Неформальная постановка задачи. В общем виде неформальная постановка задачи может быть описана следующим образом. На сцене, имеющей сложную геометрию, располагаются объекты-цели, которые необходимо контролировать с помощью средств наблюдения. Сенсоры могут располагаться в ограниченном наборе позиций внутри сцены. Препятствия на сцене, её рельеф могут создавать «слепые зоны» для средств наблюдения, причём данные зоны могут быть «слепыми» для сенсоров различного типа: аудиальных, визуальных, датчиков движения. Главной задачей является вычисление позиций сенсоров, позволяющих минимизировать размеры таких зон за счёт выбора позиций для средств наблюдения. Эта задача является двойственной по отношению к задаче максимизации «обозреваемости» сцены.

Считаем, что имеется произвольное количество объектов, которые необходимо наблюдать с помощью определённого числа камер. В этой статье мы не рассматриваем вопросы идентификации объектов наблюдения, определения траектории их движения, определения их местоположения. Мы считаем, что их сово-

купность задаётся набором возможных позиций на сцене, в частности, координатами на плоской сцене. Последние определяются, исходя из трёхмерной модели, позволяющей определить, может ли объект наблюдаться в определённой позиции камерой, находящейся в одной из фиксированных точек (рис. 1).

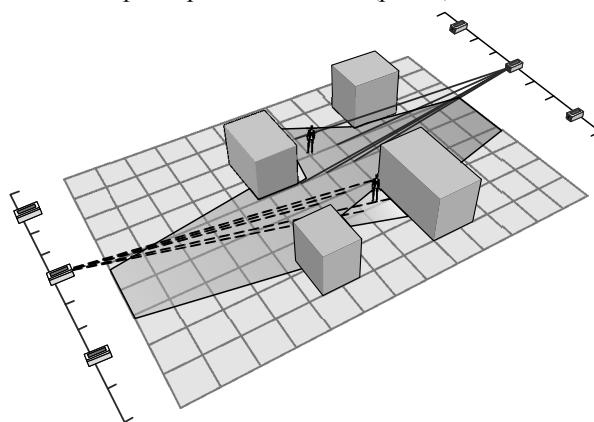


Рис. 1. Наблюдаемые объекты на трёхмерной сцене

Предполагаем, не умаляя общности, что на обозреваемую трёхмерную сцену наложена сетка (мы рассматриваем плоскую, но это не умаляет общности подхода), имеющая определённый шаг и размеры, узлы которой рассматриваются как места возможного размещения объектов-целей. На сцене имеются препятствия, под которыми узлы сетки не рассматриваются.

Такой приём часто используется в задачах оптимизации расположения камер [18]. Камеры могут находиться не в любом положении, а только в конечном множестве дискретных позиций, определяемых шагом сетки, предустановленными параметрами и ограничениями. Такая постановка позволяет рассматривать широкий класс прикладных задач оптимизации расположения камер как в закрытых помещениях, так и на достаточно обширных площадках (например, промышленных).

Далее задача поиска расположения камер сводится к оптимизационной задаче на специальном образом построенном графе, наложенном на рельеф сцены.

Формализация задачи. В процессе формализации не учитываются параметры объектов и тех предметов на сцене, которые могут формировать слепые зоны для камер наблюдения. Задача рассматривается в варианте, когда определение наблюдаемости сводится к плоской карте, на которой наблюдаемость узла определяется возможностью проведения прямой линии от сенсора к позиции узла, не проходящей через элементы-препятствия на сцене. При этом данная линия может обозначать как линию видимости, так, например, и «слышимость» данного узла аудиальным устройством. Позиция сенсора размещена в трёхмерной сетке, а позиции, за которыми необходимо вести наблюдение, – в плоской сетке.

Фактически задача оптимального расположения сенсоров решается в соответствии с определением степени наблюдаемости участков плоской карты, в которых могут располагаться объекты (рис. 2). Далее

мы предлагаем определение взвешенного графа, соответствующего описанию сцены и возможным положениям объектов и сенсоров.

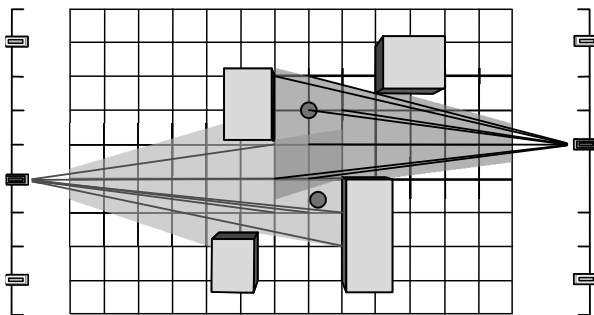


Рис. 2. Переход к «плоской» сцене

Определим ориентированный взвешенный граф наблюдаемости позиций объектов следующим образом (в этом определении предполагается наличие k сенсоров).

Графом наблюдаемости позиций объектов (далее просто граф наблюдаемости) является система

$$G = (V \cup \{A_i\}, E, f), \tag{1}$$

где V – множество вершин, соответствующее узлам сетки, возможным позициям объектов, в которых они могут располагаться, $A = \cup \{A_i\}_{i=1,k}$ – объединение набора множеств вершин, соответствующих узлам трёхмерной сетки, в которых могут располагаться сенсоры.

Каждое множество A_i является множеством вершин графа, которые соответствуют узлам сетки, в которых может располагаться i -й сенсор (каждый сенсор имеет набор параметров, их уточнение является отдельным вопросом).

E – множество рёбер графа, которое определено следующим образом. Вершина $v \in V$ соединяется с вершиной $u \in A_i$ в том и только в том случае, если узел сетки на сцене, соответствующий вершине v , обозревается соответствующим сенсором в позиции u .

Пусть $|V| = n$, $|A| = k$. То есть на сцене имеется n позиций, в которых могут располагаться объекты, и k сенсоров, которые могут находиться в вершинах, соответствующих их возможным позициям в множествах A_i .

Вершина $v \in V$ наблюдается i -м сенсором из позиции u , если $(v, u) \in E$. Принадлежность вершины u множеству A_i указывает на конкретный сенсор, позволяющий наблюдать за объектом в позиции v . Если несколько объектов находятся в одном узле нашей сетки, они будут представлены одной вершиной графа наблюдаемости G .

Для каждого ребра графа определяется его вес следующим образом:

$$f: E \rightarrow R, \text{ функция меры наблюдаемости позиции.}$$

Функция $f(e)$ принимает значение в множестве вещественных чисел. Конкретный вид функции $f(e)$ может быть определён, исходя из геометрических атрибутов сцены, экспертных оценок наблюдаемости или оценок, основанных на процедурах калибровки сенсора, параметрах освещённости сцены. В данной

работе используется вес ребра, равный 1, если вершина видна из соответствующей позиции сенсора. В более общем виде вычисление значения функции наблюдаемости может быть основано на интегральном показателе, зависящем от положения сенсора в трёхмерной сцене, фокусного расстояния камеры, вектора направления её обзора, ширины и высоты области обзора, расстояния между камерой и наблюдаемой позицией, характеристиках освещения.

Определение конфигурации сенсоров и конфигурации объектов

Множество вершин $\bar{a} = \{a_1, a_2, \dots, a_k\}$, где $\forall_{1 \leq i \leq k} a_i \in A_i$, назовём конфигурацией расположения сенсоров. Множество всех возможных конфигураций расположения сенсоров обозначаем как \bar{A} .

Назовём любое подмножество V конфигурации расположения объектов на сцене, то есть $\bar{v} = \{v_1, v_2, \dots, v_l\}$, где l – количество объектов в конфигурации. Множество всех таких конфигураций является множеством всех подмножеств V и обычно обозначается как $P(V)$.

Для любого объекта, находящегося в позиции, соответствующей вершине v , определяем наблюдаемость объекта при конфигурации сенсоров \bar{a} как функцию $F(v, \bar{a}) = \sum_{1 \leq i \leq k} f(v, a_i)$. То есть функция наблюдаемости объекта при определённой конфигурации камер является отображением следующего вида:

$$F: V \times \bar{A} \rightarrow R. \tag{2}$$

Таким образом, имеется $(k+1)$ -дольный граф, в котором могут быть соединены только вершины, соответствующие позициям наблюдаемых объектов с вершинами, соответствующими позициям сенсоров. Если позиция не наблюдается сенсором из соответствующего положения, эти две вершины не соединены ребром. Если же вершина наблюдается, существует ребро, вес которого определяется функцией наблюдаемости этой позиции на реальной сцене.

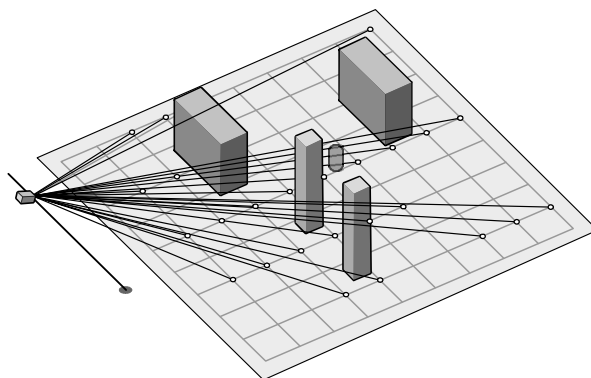


Рис. 3. Пример графа наблюдаемости для камеры в определённом положении

Целью исследования является определение и минимизация слепых зон на сцене. Минимизация достигается размещением сенсоров в некотором множестве возможных их положений, которое минимизирует ко-

личество позиций, недоступных для наблюдения. Дадим формальное определение задачи в терминах графа видимости, определённого выше.

Формализация задачи минимизации слепых зон

Слепой зоной для данной конфигурации сенсоров наблюдения $\bar{a} \in \bar{A}$ называется множество вершин $B(\bar{a}) \in V$ графа наблюдаемости G , определённого следующим образом:

$$B(\bar{a}) = \{ v \in V \mid F(v, \bar{a}) = 0 \}. \tag{3}$$

Задача минимизации слепых зон заключается в поиске конфигурации камер наблюдения, для которых количество узлов сетки вне зоны наблюдения будет минимальным:

$$|B(\bar{a})| \rightarrow \min_{\bar{a} \in \bar{A}}. \tag{4}$$

Псевдокод алгоритма определения слепых зон в трёхмерной динамической сцене

По определению графа наблюдаемости вершина $v \in V$ сцены принадлежит слепой зоне для данной конфигурации \bar{a} только в том случае, когда $F(v, \bar{a}) = 0$. То есть для всех $a_i \in \bar{a}$ должно выполняться условие:

$$|\{v \mid (v, a_i) \in E\}| = 0. \tag{5}$$

По этой причине при вычислении размера слепой зоны для заданной конфигурации используется алгоритм, который не выполняет проверку видимости всех позиций сцены для всех сенсоров. Значительное сокращение времени работы алгоритма происходит за счёт того, что он осуществляет проверку всех позиций сцены только для одного сенсора (выбирается исходя из их экспертного ранжирования или случайным образом). Для остальных полученная при первом прогоне слепая зона «сужается», из неё удаляются те позиции v , для которых функция меры наблюдаемости f не равна нулю, как это должно быть в соответствии с (5).

В общем случае при полной проверке видимости всех позиций для каждого сенсора сложность алгоритма поиска слепой зоны для одной конфигурации имеет порядок $n \times k$, где k – количество сенсоров наблюдения, $n = |V|$. То есть зависимость имеет линейный вид от k . Результаты и особенности вычислительного эксперимента описаны далее.

Описание алгоритма

Входные данные: Граф $G = (V \cup \{A_i\}, E, f)$, конфигурация сенсоров слежения \bar{a} .

Выходные данные: количество вершин N в слепой зоне для конфигурации.

Описанная ниже функция возвращает число вершин графа наблюдаемости G , которые входят в слепую зону при конфигурации \bar{a} . Для хранения вершин графа, которые входят в слепую зону, используется структура данных «список».

Function GetBlindArea(G, \bar{a}): **Integer**

```

N:=|V|
//Текущие вершины в слепой зоне
BlindNodeList.Create
//Добавляем все вершины сцены в слепую зону
for всех вершин  $v \in V$  do BlindNodeList.Add(v)
for всех сенсоров  $a$  конфигурации  $\bar{a}$  do begin
  if Not BlindNodeList.IsEmpty then
    for всех вершин  $v \in$  BlindNodeList do begin
      //удаляем вершину  $v$  из слепой зоны, если она видима
      if  $(v, a) \in E$  then BlindNodeList.DeleteNode(v)
    end
  end
N:=BlindNodeList.Size
Return(N)
end

```

Структура программного комплекса. Для моделирования трёхмерной сцены и решения задачи оптимизации размещения сенсоров разработан программный комплекс, который позволяет строить и в реальном времени обновлять граф наблюдаемости, а также решать формализованную задачу (предусмотрены варианты получения оптимального решения или приближённого решения, полученного эвристическими алгоритмами).

Инструментарий позволяет строить виртуальную сцену большой сложности и содержит следующие типы объектов:

- **Цель.** Представляет собой объект, расположенный на сцене, установление наблюдения за которым и является целью данного комплекса.
- **Препятствия.** Имитируют реальные препятствия на сцене, мешают установлению визуального контакта камеры с объектом-целью. Могут иметь геометрическую форму различной сложности.
- **Средства наблюдения.** Могут иметь различные алгоритмы работы, основываясь на различных функциях наблюдения.

Программный комплекс содержит в себе три группы классов.

Классы описания – содержат описание сущностей и объектов сцены (рис. 4).

Данная группа классов содержит:

- Классы для описания и построения сетки сцены. Сетка сцены предназначена для последующего формирования графа всей сцены и представляет собой набор узлов, расположение в пространстве которых зависит от способа построения сетки. На текущий момент реализована сетка «таблица», узлы которой в каждой из трёх плоскостей равноудалены от своих соседей, а также сетка «проекция», в которой узлы спроецированы на поверхность сложного объекта (данная сетка предназначена для использования на неровном ландшафте).
- Классы для описания сцены и её объектов:
 - Класс *Scene* позволяет строить необходимую нам сцену, а также управлять объектами на ней.

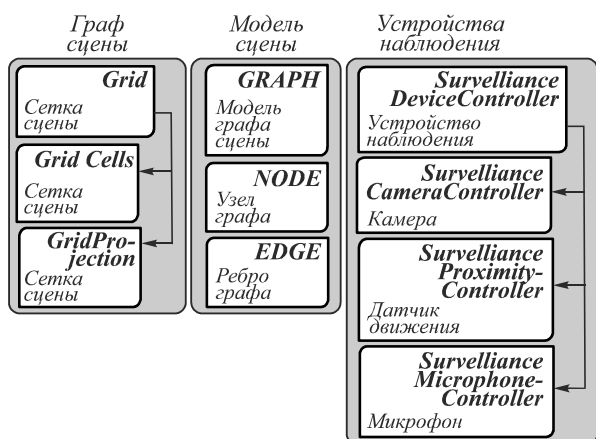


Рис. 4. Структура основных классов комплекса

- **Класс Scene Object** – главный родительский класс для всех объектов сцены. Позволяет производить «привязку» объекта к узлам сетки (определять, к какому из узлов принадлежит объект).
- **SurveillanceDeviceController.** Базовый класс для всех сенсоров. Объекты класса имеют общие для всех видов сенсоров характеристики – пространственное положение, методы получения узлов сетки, которые «наблюдаются» данным устройством. Наследниками этого класса являются: *SurveillanceCameraController*, *SurveillanceMicrophoneController* и *SurveillanceProximityController*, которые описывают логику работы сенсоров, соответственно, камеры, микрофона и датчика перемещения. Все сенсоры имеют характеристики реальных устройств, учитываемые при определении наблюдаемости узлов этим сенсором.
- Классы, используемые для построения графа видимости.
 - Класс *Graph* описывает граф видимости и используется при построении модели сцены.
 - Классы *Node* и *Edge* используются для описания узлов и рёбер соответственно. Узлы хранят собственное положение в пространстве и список смежных рёбер, имеющих определённый вес.

Классы логики предназначены для проведения вычислений и описывают алгоритм оптимизации для решения поставленной задачи определения оптимального расположения камер в пространстве.

- **Computation Center** содержит алгоритмы вычисления параметров и свойств графа (объект класса *Graph*).
- Статический класс *Evaluation Function* содержит оценочные функции видимости объекта, которые используются при формировании графа, а именно при определении весов рёбер, соединяющих сенсоры и цели.
- Класс *Graph Builder* позволяет производить построение взвешенного ориентированного графа с применением имеющейся сетки сцены (*Grid*) и объектов сцены (*Scene Object*).

Алгоритмы вычислений, связанных с решением задачи вычисления слепой зоны для данной конфигурации сенсоров, размещаются в наследниках класса *ComputationModuleController* (рис. 5). Данные классы также предназначены для вывода на экран интерфейса управления этими модулями.

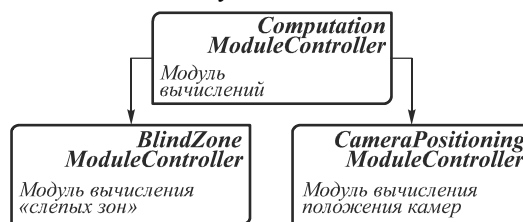


Рис. 5. Классы модулей вычисления слепых зон и положений камер наблюдения

Классы визуализации (рис. 6) исполняют единственную функцию – создания графического представления имеющейся сцены, а также её модели.

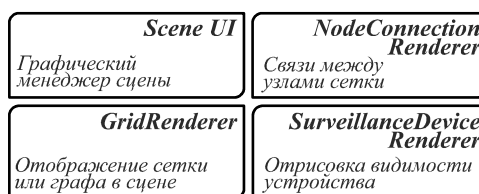


Рис. 6. Классы логики и визуализации

- Главный класс *Scene UI* для управления графическим представлением сцены и обеспечением взаимодействия с пользователем.
- Класс отрисовки *GridRenderer*, позволяющий создавать видимое представление графа и сетки сцены. Для отображения узлов и связей между ними данный класс использует вспомогательный класс *Node Connection Renderer*.
- Вывод на экран визуального представления связи камеры с сеткой или графом осуществляется посредством класса *SurveillanceDeviceRenderer*, отображающего связи между камерой и видимыми узлами сетки сцены.

Применение

Программный комплекс позволяет оптимизировать расположение сенсоров в трёхмерной сцене с целью минимизации «слепых зон» (рис. 7). При вычислении областей наблюдения сенсоров учитываются установленные параметры средств наблюдения (в случае камер – их угол обзора, дальность видимости, фокусное расстояние), что позволяет добиться возможности тонкой настройки и высокой точности расчётов.

При выполнении расчётов вычисляются дополнительные параметры как самой сцены, так и самого процесса обработки данных: процент покрытия площади, процент перекрытия областей воздействия сенсоров, время расчёта текущей конфигурации.

Результаты вычислительного эксперимента

Для оценки скорости обработки сцены с целью поиска конфигурации сенсоров с минимальной по размеру слепой зоной был проведён вычислительный

эксперимент, который выполнялся на ноутбуке с процессором Intel Core i5-6300HQ, с 8GB DDR3 SDRAM.

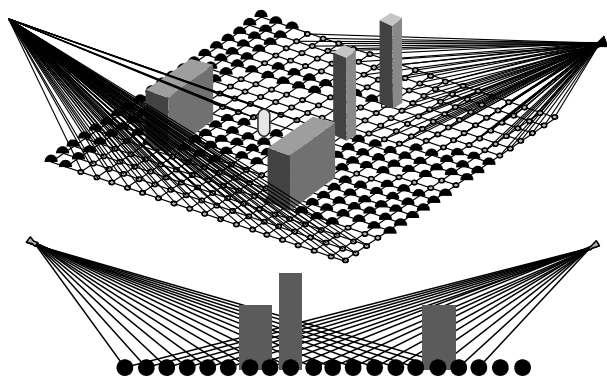


Рис. 7. Применение двух сенсоров наблюдения для определения «слепых зон» сцены

Моделирование сцены и расчёт осуществлялись с помощью программного обеспечения Unity 3D 5.4.1. В качестве сцены, которая моделировалась в эксперименте, рассматривалась модель плоской сцены с различным количеством случайно размещённых объектов-препятствий и сенсоров. Видимость объекта определяется с помощью стандартной функции RayCast, реализованной в используемом программном обеспечении.

Дадим общую формулу вычисления времени работы алгоритма поиска минимальной слепой зоны для заданной сцены и количества камер наблюдения. Введём следующие обозначения:

- T_1 – время сканирования для одного сенсора сцены, на которой находится один объект-препятствие;
- T_2 – время, на которое увеличивается время сканирования одной конфигурации при добавлении одного объекта-препятствия;
- m – количество объектов препятствий, которые формируют граф видимости;
- k – количество используемых сенсоров.

На рис. 8 приведены графики характера временной сложности алгоритма вычисления слепой зоны для заданной конфигурации сенсоров с помощью полной проверки всех позиций для каждого сенсора (линейная функция) и с помощью алгоритма **GetBlindArea**. На графике показано, как растёт вычислительная сложность в том и в другом случае при увеличении числа препятствий m , в частности при $m = 10$. Вычисления производились для сгенерированных случайным образом сцен, которые имеют физические размеры порядка 10 тыс. м² с шагом сетки, равным 1 метру.

Вычислительный эксперимент показал, что поиск слепой зоны для заданной конфигурации со случайным расположением препятствий по времени зависит от количества сенсоров (k) с видом, близким к логарифмической зависимости $T_1 + \log(1 + T_1 \times (k - 1) \times \alpha)$, где $T_1 = 14$ мс для нашей сцены. Величина параметра α определяется в ходе вычислительного эксперимента, его оптимальное значение, наиболее точно соответствующее полученным результатам, было определено эмпирическим путём и равно 2. Каждое новое препятствие увеличивает время обчёта всей сцены для одной камеры на время $T_2 = 0,01$ мс.

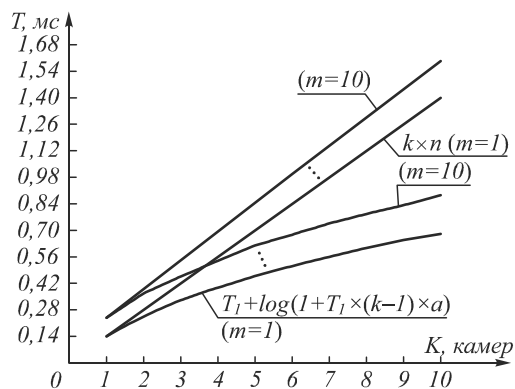


Рис. 8. Время выполнения проверок RayCast

Для решения же полной задачи необходимо перебрать все возможные конфигурации сенсоров, что требует числа операций порядка

$$(T_1 + \log(1 + T_1 \times (k - 1) \times \alpha) + T_2 \times (m - 1) \times k) \times \prod_{i=1}^k |A_i|. \quad (6)$$

Существуют конфигурации сенсоров и объектов-препятствий, которые могут в худшем случае привести к количеству операций, растущему линейным образом, но при генерации случайных позиций и геометрических размеров препятствий на сцене при усреднении результатов вычислений справедлива логарифмическая зависимость, как это показано на рис. 8.

Известно, что задача оптимизации расположения камер в общем случае является *NP-полной* (см. об этом в [6]). Такая оценка достигается за счёт экспоненциального роста количества комбинаций расположения сенсоров (последний множитель в формуле 6). Предложенный алгоритм позволяет сократить время обчёта одной конфигурации и тем самым уменьшить время решения всей задачи поиска оптимального расположения сенсоров. Для сокращения сложности задачи необходимо использовать экспертные оценки возможных положений сенсоров, что позволяет за приемлемое время получить оптимальное решение.

Заключение

В статье представлена формализация для задачи определения оптимального расположения камер наблюдения с помощью специального графа видимости. Для этого графа сформулирована задача нахождения слепых зон. Предложен алгоритм их поиска, который по результатам вычислительного эксперимента существенно сокращает трудоемкость для одной конфигурации сенсоров и конфигурации объектов-препятствий, что позволяет решить задачу оптимального размещения сенсоров с целью максимального покрытия сцены. Описана архитектура программного комплекса, позволяющего моделировать размещение объектов и сенсоров в трёхмерной сцене, реализованного в среде Unity 3D 5.4.1. Проведён вычислительный эксперимент, и определены временные параметры работы алгоритма расчёта одной конфигурации сенсоров, используемого для поиска оптимального их расположения, минимизирующего слепые зоны на сцене.

Литература

1. **Давидюк, Н.В.** Автоматизация процедуры подбора технических средств обнаружения системы физической защиты объектов / Н.В. Давидюк // Вестник Архангельского ГТУ. Серия: Управление, вычислительная техника и информатика. – 2009. – № 1. – С. 98-100.
2. **El-Attar, A.** A robust multistage algorithm for camera self-calibration dealing with varying intrinsic parameters / A. El-Attar, M. Karim, H. Tairi, S. Ionita // Journal of Theoretical and Applied Information Technology. – 2011. – Vol. 32, No. 1. – P. 46-54.
3. **Civera, J.** Camera self-calibration for sequential bayesian structure from motion / J. Civera, D.R. Bueno, A.J. Davison, J.M.M. Montiel // In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA'09). – 2009. – P. 403-408. – DOI: 10.1109/ROBOT.2009.5152719.
4. **Fiore, L.** Optimal camera placement with adaptation to dynamic scenes / L. Fiore, G. Somasundaram, A. Drenner, N. Papanikolopoulos // In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA 2008). – 2008. – P. 956-961. – DOI: 10.1109/ROBOT.2008.4543328.
5. **Bodor, R.** Multi-camera positioning to optimize task observability / R. Bodor, P. Schrater, N. Papanikolopoulos // In: Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS 2005). – 2005. – P. 552-557. – DOI: 10.1109/AVSS.2005.1577328.
6. **Zhao, J.** Approximate techniques in solving optimal camera placement problems / J. Zhao, D. Haws, R. Yoshida, S. Cheung // International Journal of Distributed Sensor Networks. – 2013. – Vol. 2013. – P. 241913. – DOI: 10.1155/2013/241913.
7. **Hänel, M.** Optimal camera placement to measure distances conservatively regarding static and dynamic obstacles / M. Hänel, S. Kuhn, D. Henrich, J. Pannek, L. Grüne // International Journal of Sensor Networks. – 2012. – Vol. 12, Issue 1. – P. 25-36. – DOI: 10.1504/IJSNET.2012.047713.
8. **Kim, H.** Dynamic 3D scene reconstruction in outdoor environments / H. Kim, M. Sarim, T. Takai, J.-Y. Guillemaut, A. Hilton // Proceedings of International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT). – 2010. – P. 613-626.
9. **Liu, J.** Automatic camera calibration and scene reconstruction with scale-invariant features / J. Liu, R. Hubbard // Proceedings of the Second international Symposium: Advances in Visual Computing (ISVC'06). – 2006. – Part 1. – P. 558-568. – DOI: 10.1007/11919476_56.
10. **Van den Hengel, A.** Automatic camera placement for large scale surveillance networks / A. Van den Hengel, R. Hill, B. Ward, A. Cichowski, H. Detmold, C. Madden, A. Dick, J. Bastian // Proceedings of WACV. – 2009. – P. 1-6. – DOI: 10.1109/WACV.2009.5403076.
11. **Пат. RU 2561925 Российская Федерация А 01 G 23/00.** Способ определения оптимальной конфигурации системы видеомониторинга леса / Шишалов И.С., Филимонов А.В., Громазин О.А., Пархачев В.В.; заявитель и правообладатель Общество с ограниченной ответственностью "ДиСиКон"; опубл. 25.09.2015 г.
12. **Holt, R.** Summary of results on optimal camera placement for boundary monitoring / R. Holt, M. Hong, R. Martini, I. Mukherjee, R. Netravali, J. Wang // Proceedings of SPIE. – 2007. – Vol. 6570. – P. 657005. – DOI: 10.1117/12.719139.
13. **Ажмухамедов, И.М.** Формализация задачи размещения элементов охранной системы в контролируемой зоне / И.М. Ажмухамедов // Вестник Астраханского государственного технического университета. – 2008. – № 1(42). – С. 77-79.
14. **Van Beeck, K.** Real-time vision-based pedestrian detection in a truck's blind spot zone using a warping window approach / K. Van Beeck, T. Goedemé, T. Tuytelaars // In: Informatics in Control, Automation and Robotics / ed. by J. Ferrier, A. Bernard, O. Gusikhin, K. Madani. – Chap. 16. – Berlin: Springer International Publishing, 2014. – P. 251-264. – DOI: 10.1007/978-3-319-03500-0_16.
15. **Cardarelli, E.** Vision-based blind spot monitoring / E. Cardarelli // Handbook of Intelligent Vehicles. – 2012. – P. 1071-1087. – DOI: 10.1007/978-0-85729-085-4_44.
16. **Печенкин, В.В.** Проектный подход к формированию IT-компетенций технических специалистов в рамках разработки программного комплекса для оптимизации размещения камер наблюдения за объектами / В.В. Печенкин, Д.А. Лепесткин // В кн.: Современные методы преподавания для студентов инженерных направлений: монография / под общ. ред. О.Н. Долиной. – Саратов: Саратовский государственный технический университет, 2014. – С. 156-168.
17. **Печенкин, В.В.** Построение модели управления в сложных динамических технических системах / В.В. Печенкин, Д.С. Решетников // В сб.: Проблемы управления в социально-экономических и технических системах. Сборник научных статей по материалам X Всероссийской научной конференции 10-11 апреля 2014 года. Саратов: СГТУ, 2014. – С. 16-19.
18. **Yusoff, S.K.M.** Optimal camera placement for 3D environment / S.K.M. Yusoff, A.Md Said, I. Ismail // Proceedings of the International Conference on Software Engineering and Computer Systems (ICSECS 2011). – Part II. – P. 448-459. – DOI: 10.1007/978-3-642-22191-0_39.

Сведения об авторах

Печенкин Виталий Владимирович, 1956 года рождения, в 1978 году окончил Саратовский государственный университет по специальности «Прикладная математика», работает профессором кафедры прикладных информационных технологий. Область научных интересов: оптимизационные задачи на графах и сетях, визуализация графов, методы статистической обработки многомерных данных. E-mail: pechenkinvv@sstu.ru.

Королёв Михаил Сергеевич, 1992 года рождения, в 2015 году окончил Саратовский государственный технический университет по специальности 230204 «Информационные технологии в медиаиндустрии», работает ассистентом кафедры прикладных информационных технологий в СГТУ имени Гагарина Ю.А. Область научных интересов: виртуальная реальность, дополненная реальность, 3D-проектирование, 3D-моделирование, архитектурная визуализация, компьютерная оптика, обработка изображений, компьютерный дизайн, цифровая фотография. E-mail: koroliow.mikhail@yandex.ru.

ГРПТИ: 28.17.33 .

Поступила в редакцию 3 июня 2016 г. Окончательный вариант – 21 марта 2017 г.

OPTIMAL PLACEMENT OF SURVEILLANCE DEVICES IN A THREE-DIMENSIONAL ENVIRONMENT FOR BLIND ZONE MINIMIZATION

V.V. Pechenkin¹, M.S. Korolev¹.

¹Yu. A. Gagarin Saratov State Technical University, Saratov, Russia

Abstract

This paper discusses the usage of devices of various types and configuration in the developed software for blind zone minimization via optimal placement of the surveillance devices when observing targets in a complex three-dimensional dynamic scene. We describe the architecture of the software complex, the principle of operation of the surveillance devices, and a blind zone detection algorithm.

We propose a formalization of the problem, also describing specific tasks of object visibility definition, reduced to solving the optimization problems of different complexity classes based on a special visibility graph.

Keywords: surveillance devices, audio sensors, video sensors, placement optimization, visibility, observability.

Citation: Pechenkin VV, Korolev MS. Optimal placement of surveillance devices in a three-dimensional environment for blind zone minimization. *Computer Optics* 2017; 41(2): 245-253. DOI: 10.18287/2412-6179-2017-41-2-245-253.

References

- [1] Davidyuk NV. Automation of detecting sensors selection procedure in physical protection systems. *Vestn Astrakhan State Technical Univ. Ser Management, Computer Sciences and Informatics* 2009; 1: 98-100.
- [2] El-Attar A, Karim M, Tairi H, Ionita S. A robust multi-stage algorithm for camera self-calibration dealing with varying intrinsic parameters. *Journal of Theoretical and Applied Information Technology* 2011; 32(1): 46-54.
- [3] Civera J, Bueno DR, Davison AJ, Montiel JMM. Camera self-calibration for sequential bayesian structure from motion // In: *Proc ICRA'09* 2009; 403-408. DOI: 10.1109/ROBOT.2009.5152719.
- [4] Fiore L, Somasundaram G, Drenner A, Papanikolopoulos N. Optimal camera placement with adaptation to dynamic scenes. In: *Proc ICRA'08* 2008; 956-961. DOI: 10.1109/ROBOT.2008.4543328.
- [5] Bodor R, Schrater P, Papanikolopoulos N. Multi-camera positioning to optimize task observability // In: *Proc AVSS* 2005; 552-557. DOI: 10.1109/AVSS.2005.1577328.
- [6] Zhao J, Haws D, Yoshida R, Cheung S. Approximate techniques in solving optimal camera placement problems. *International Journal of Distributed Sensor Networks* 2013; 2013: 241913. DOI: 10.1155/2013/241913.
- [7] Hänel M, Kuhn S, Henrich D, Pannek J, Grüne L. Optimal camera placement to measure distances conservatively regarding static and dynamic obstacles. *International Journal of Sensor Networks* 2012; 12(1): 25-36. DOI: 10.1504/IJSNET.2012.047713.
- [8] Kim H, Sarim M, Takai T, Guillemaut J-Y, Hilton A. Dynamic 3D scene reconstruction in outdoor environments. *3DPVT* 2010; 613-626.
- [9] Liu J, Hubbold R. Automatic Camera Calibration and Scene Reconstruction with Scale-Invariant Features. *ISVC'06* 2006; 1: 558-568.
- [10] Hengel VD, Hill R, Ward B, Cichowski A, Detmold H, Madden C, Dick A, Bastian J. Automatic camera placement for large scale surveillance networks. *Proc WACV* 2009; 1-6. DOI: 10.1109/WACV.2009.5403076.
- [11] Shishalov IS, Filimonov AV, Gromazin OA, Parkhachev VV. Method for determining the optimal configuration of the video monitoring system of the forest [In Russian]. *Pat RF of Invent N 2561925* of September 25, 2015.
- [12] Holt R, Hong M, Martini R, Mukherjee I, Netravali R, Wang J. Summary of results on optimal camera placement for boundary monitoring. *Proc SPIE* 2007; 6570: 657005. DOI: 10.1117/12.719139.
- [13] Azhmukhamedov IM. Formalization of the task of placing elements of the security system in the controlled area [In Russian]. *Vestnik of Astrakhan State Technical University* 2008; 1(42): 77-79 .
- [14] Van Beeck K, Goedemé T, Tuytelaars T. Real-time vision-based pedestrian detection in a truck's blind spot zone using a warping window approach. In: *Ferrier J, Bernard A, Gusikhin O, Madani K, eds. Informatics in Control, Automation and Robotics. Chapter 16. Berlin: Springer International Publishing; 2014: 251-264. DOI: 10.1007/978-3-319-03500-0_16.*
- [15] Cardarelli E. Vision-based blind spot monitoring. *Handbook of Intelligent Vehicles* 2012: 1071-1087. DOI: 10.1007/978-0-85729-085-4_44.
- [16] Pechenkin VV, Lepestkin DA. The project approach to the formation of IT-competencies of technical specialists in the development of a software package for optimizing the location of surveillance cameras for objects [In Russian]. In *Book: Dolinina ON, ed. Modern methods of teaching for engineering students [In Russian]. Saratov: State Technical University of Saratov Publisher; 2014: 156-168.*
- [17] Pechenkin VV, Reshetnikov DS. Building a model of governance in complex dynamic technical systems [In Russian]. In *Book: Management problems in socio-economic and technical systems. Saratov: "SGTU" Publisher, 2014: 16-19.*
- [18] Yusoff SKM, Md Said A, Ismail I. Optimal camera placement for 3D environment. *Proc ICSECS* 2011; II: 448-459. DOI: 10.1007/978-3-642-22191-0_39.

Authors' information

Vitaliy Vladimirovich Pechenkin (b. 1956) graduated from Saratov State University in 1978, majoring in Applied Mathematics. He works as a professor of the International School of Applied Information Technology. Research interests are optimization problems on graphs and networks, graph visualization, statistical treatment of multidimensional data. E-mail: pechenkinvv@sstu.ru.

Mikhail Sergeevich Korolev (b. 1992) graduated from Saratov State Technical University in 2015 majoring in Information Technology at Multimedia. He works as an assistant of of the International School of Applied Information Technology. Research interests are virtual reality, augmented reality, 3d-technology, computer optics, image processing, computer design, digital photography. E-mail: koroliow.mikhail@yandex.ru.

Received June 3, 2016. The final version – March 21, 2017.
