

СИСТЕМА НАВИГАЦИИ НА ВЫСОКО РЕАЛИСТИЧНОЙ МОДЕЛИ МЕЖДУНАРОДНОЙ КОСМИЧЕСКОЙ СТАНЦИИ

А.В. МАЛЬЦЕВ^{*}, М.В. МИХАЙЛЮК[†]

^{*} Научно-исследовательский институт системных исследований РАН
Москва, Россия
e-mail: avmaltcev@mail.ru

[†] Научно-исследовательский институт системных исследований РАН
Москва, Россия
Email: mix@niisi.ras.ru

Ключевые слова: Виртуальная реальность, навигация, граф, путь, видеотренажеры, высоко реалистичные модели

Аннотация. Выходя на поверхность Международной космической станции (МКС), космонавт должен передвигаться, держась за специальные поручни. Поскольку МКС имеет большие размеры и, передвигаясь по ней, можно просто заблудиться, то для эффективного и безопасного перемещения необходимо решать задачу навигации, состоящую в построении кратчайшего маршрута (проходящего по поручням) от начального положения до целевой точки. В статье предлагаются алгоритмы поиска кратчайших маршрутов между двумя заданными поручнями, которые могут применяться для реализации системы навигации в космических тренажерных комплексах подготовки космонавтов к работе на МКС.

1 ВВЕДЕНИЕ

В качестве одного из крупнейших направлений деятельности в сфере изучения космоса, а также процессов и явлений, происходящих в нем, выступает Международная космическая станция (МКС). Она позволяет проводить научные и технические эксперименты в различных областях знаний. Подготовка специалистов для работы на МКС осуществляется с помощью специальных тренажерных комплексов и систем, основанных на использовании технологий виртуальной реальности. В таких комплексах и системах реальная обстановка (как внутри, так и вне станции) заменяется на визуализируемую трехмерную виртуальную сцену¹. На рисунке 1 приведен пример такой сцены, включающей высоко реалистичную модель МКС. Эта модель содержит более 1 млн. полигонов и более 358 МБ текстур (включая текстуры рельефа, отражения и прозрачности). При ее визуализации с помощью системы GLView² моделируются источники освещения типа «фара», а также мягкие и жесткие тени.

Одной из важных и актуальных задач, возникающих при обучении космонавтов, является формирование навыков ориентации на внешней поверхности МКС. В настоящее время станция насчитывает несколько десятков модулей и имеет протяженность более ста метров. Передвигаясь по ней, космонавт может просто заблудиться, поэтому навыки ориентации очень важны для его эффективной и

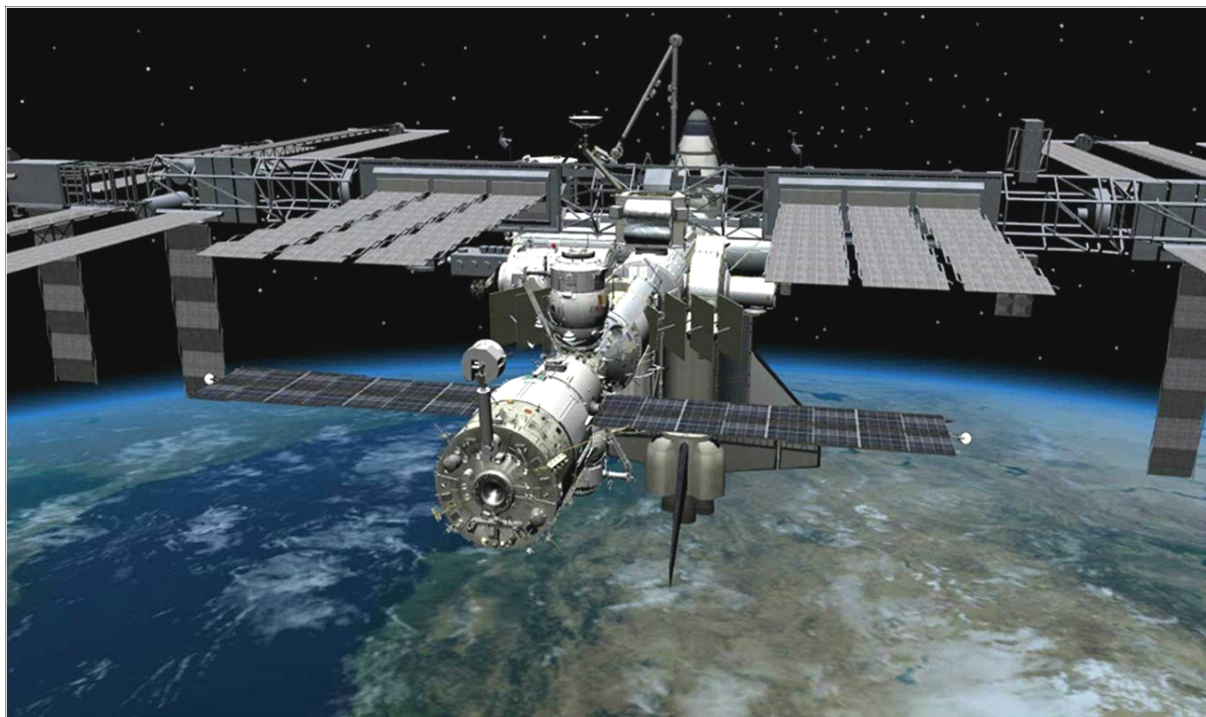


Рис. 1: Модель Международной космической станции.

безопасной работы. Перемещение по внешней стороне станции от одного объекта к другому необходимо осуществлять, держась за специальные поручни (рис. 2), которые закреплены на поверхности космических модулей. Это требование связано с обеспечением безопасности при работе в условиях открытого космоса. Для эффективного передвижения от начального положения к нужному объекту желательно знать кратчайший маршрут, то есть маршрут, содержащий наименьшее количество поручней.

Прежде, чем выйти в открытый космос и проводить работы в удаленных частях станции, космонавт должен выработать маршрут передвижения по поручням от его начального положения (например, выходного люка) до места назначения (места проведения необходимых работ). Этот маршрут он может выделить на виртуальной модели МКС, отмечая ручную поручни пути с помощью указателя или используя автоматические алгоритмы. В данной статье предлагаются алгоритмы поиска кратчайших



Рис. 2: Перемещение космонавта по поручням.

маршрутов между двумя заданными поручнями, позволяющие реализовать навигационную систему в космическом видеотренажере.

2 ПОИСК КРАТЧАЙШЕГО МАРШРУТА

Для определения кратчайшего маршрута между двумя поручнями предлагается использовать неориентированный граф^{3,4} G , вершины которого соответствуют объектам типа «поручень» в виртуальной модели станции, применяемой в имитационно-тренажерных комплексах подготовки космонавтов. При этом любые две вершины A и B графа связаны ребром, если космонавт может непосредственно переместиться с поручня A на поручень B .

Основой предлагаемого алгоритма нахождения кратчайшего маршрута в графе G от начальной вершины V_H до целевой вершины V_K (а значит, и кратчайшего маршрута между соответствующими поручнями на поверхности станции) является остовное дерево^{5,6} G' графа G с корнем в V_K (т.е. дерево, которое содержит все вершины графа G). Заметим, что корнем G' является конечная вершина V_K пути (а не начальная вершина V_H), так как это облегчает построение пути из V_H в V_K . Мы будем строить некоторое поддерево G'' остовного дерева G' , содержащее вершины V_K и V_H . При построении и сохранении дерева G'' в памяти компьютера будем использовать структуру данных уже имеющегося графа G . Для этого к каждой вершине G добавим указатель P на ее родительскую вершину в дереве G'' .

После того, как остовное поддерево G'' найдено, в нем в вершине V_H будет записана ссылка на родителя этой вершины, в этом родителе – ссылка на его родителя и так далее до вершины V_K . Поэтому остается только пройти по всем этим ссылкам от вершины V_H до V_K и записать индексы посещенных вершин. Эти вершины и будут входить в искомый кратчайший маршрут от V_H до V_K .

Для хранения графа G мы будем использовать файл со следующей структурой:

$$\begin{array}{l} n \\ S_0, i_0, j_{0,0}, \dots, j_{0,m} \\ \dots \\ S_{n-1}, i_{n-1}, j_{n-1,0}, \dots, j_{n-1,q} \end{array}$$

Первая строка содержит общее количество n вершин графа G . Каждая следующая строка включает последовательно символьное имя S_k ($k \in [0, n-1]$) объекта-поручня в виртуальной сцене, номер i_k соответствующей ему вершины графа G и список номеров вершин, связанных ребром с вершиной i_k .

Для работы с графом будем загружать его из файла в структуру данных, находящуюся в оперативной памяти и представляющую собой массив вершин, каждая из которых содержит:

- имя S соответствующего вершине поручня в сцене;
- список J вершин, связанных ребром с данной вершиной в графе G ;
- флаг F посещения;
- ссылку P на родительскую вершину (необходимую для построения G'').

При загрузке данных из файла флаг посещения и ссылку на родительскую вершину устанавливаем равными 0.

Рассмотрим нахождение кратчайшего маршрута между вершинами V_H и V_K графа G . Пусть $F(V)$ – флаг посещения вершины V , а $J(V)$ и $P(V)$ – список связанных с ней вершин и ссылка на родителя соответственно. Тогда получаем следующий

Алгоритм поиска кратчайшего маршрута между вершинами V_H и V_K графа

1. Организация пустой очереди Q вершин;
2. Добавление V_K в Q и установка $F(V_K)$ в 1;
3. Цикл пока Q не пуста или не найдена вершина V_H :

Выборка вершины V из Q ;

Цикл по вершинам $V_k \in J(V)$:

Если $F(V_k) = 0$, то:

$P(V_k) = V$;

Если $V_k = V_H$, то выход из цикла;

$F(V_k) = 1$;

Добавление V_k в Q ;

Конец цикла

Конец цикла

4. Проход по ссылкам P от V_H до V_K с записью в память индексов вершин, входящих в маршрут.

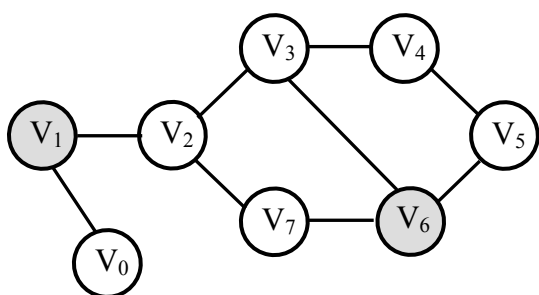


Рис. 3: Граф G .

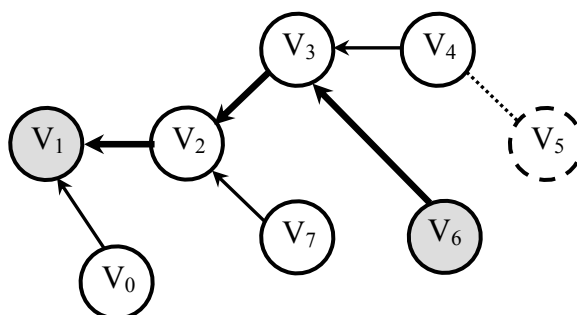


Рис. 4: Граф G'' .

Проиллюстрируем работу предложенного алгоритма с помощью графа G , изображенного на рис. 3. Допустим, что необходимо построить кратчайший маршрут из вершины V_6 в вершину V_1 . Из рисунка видно, что G содержит три пути (без циклов) из V_6 в V_1 , при этом два из них ($V_6 \rightarrow V_3 \rightarrow V_2 \rightarrow V_1$ и $V_6 \rightarrow V_7 \rightarrow V_2 \rightarrow V_1$) включают по 3 ребра и один ($V_6 \rightarrow V_5 \rightarrow V_4 \rightarrow V_3 \rightarrow V_2 \rightarrow V_1$) – 5 ребер.

После выполнения шагов 1-3 алгоритма мы получаем в памяти дерево кратчайших путей G'' , представленное на рис. 4. Поскольку вершина V_5 находится на более глубоком уровне дерева G'' от корня V_1 , чем искомая вершина V_6 , то построение G'' прекращается до ее обработки. Следовательно, V_5 не входит в G'' . Проходя по ссылкам $P(V_6)$, $P(V_3)$, $P(V_2)$ и записывая индексы посещенных вершин, получаем кратчайший маршрут из V_6 в V_1 : $V_6 \rightarrow V_3 \rightarrow V_2 \rightarrow V_1$.

Видно, что полученный маршрут совпадает с одним из двух имеющихся путей, содержащих минимальное число ребер. Выбор одного конкретного кратчайшего маршрута из нескольких возможных вариантов обусловлен порядком обработки

вершин при построении остовного поддерева. В рассмотренном примере из двух вершин одного уровня глубины от корня приоритет в порядке обработки отдавался вершине с меньшим индексом.

3 ПОИСК ВСЕХ КРАТЧАЙШИХ МАРШРУТОВ

Во многих случаях необходимо знать не просто один кратчайший маршрут между двумя поручнями, а иметь выбор из различных вариантов кратчайших маршрутов. Для этого желательно знать все возможные кратчайшие маршруты между двумя вершинами графа. Использование алгоритма, предложенного в п. 2, не может обеспечить решение данной задачи. Однако некоторая его модификация позволит выполнить поиск всех кратчайших путей между двумя вершинами графа G , а значит, и между соответствующими поручнями на поверхности станции.

Суть такой модификации алгоритма заключается в том, что вместо остовного дерева G' с корнем в V_K , мы будем строить ориентированный подграф G_1 , в котором любой путь от V_H до V_K будет кратчайшим путем между теми же вершинами в графе G . Фактически, в G_1 для каждой пары вершин все пути между ними будут иметь одинаковую длину. Формирование структуры G_1 необходимо начинать с вершины V_K . Пусть $L(V)$ – уровень вершины V в графе G_1 , равный количеству ребер на пути от V до V_K . Для каждой вершины $V \neq V_K$, будем сохранять ссылки на всех родителей $V_{P,i}$ в графе G_1 , для которых $L(V_{P,i}) = L(V) - 1$, то есть уровень родителя на единицу меньше уровня V . С целью оптимизации, достаточно ограничиться построением подграфа G_1' , содержащего вершины с уровнями до $L(V_H)$ включительно. Построив G_1' , необходимо выполнить рекурсивную функцию, проходящую по всем путям от V_H до V_K в G_1' и записывающую в память индексы вершин, входящих в эти пути.

Для реализации описанной модификации алгоритма поиска кратчайших путей определим структуру информации о каждой вершине следующим образом:

- имя S соответствующего вершине поручня в сцене;
- список J вершин, связанных ребром с данной вершиной в графе G ;
- флаг F посещения;
- уровень L вершины;
- массив ссылок P на родительские вершины.

Рассмотрим нахождение всех кратчайших маршрутов между вершинами V_H и V_K графа G . Пусть $P(V)$ – массив ссылок на родителей вершины V , $P(V)[i]$ – i -ый родитель V ; LS – конечный уровень, при достижении которого алгоритм должен завершить работу (вначале равен большому положительному числу); $size(A)$ – число элементов в массиве A ; M – массив кратчайших путей, $M[k]$ – k -ый кратчайший путь, содержащий индексы входящих в него вершин. Тогда получаем следующий

Алгоритм поиска всех кратчайших путей между вершинами V_H и V_K в графе G

1. Организация пустой очереди Q вершин;
2. Добавление V_K в Q ;
3. $F(V_K) = 1$, $L(V_K) = 0$, $LS = 1000000$;
4. Цикл пока Q не пуста:
 - Выборка вершины V из Q ;
 - Если $L(V) = LS$, то выход из цикла;

```

Цикл по вершинам  $V_k \in J(V)$ :
  Если  $F(V_k) = 0$ , то: // вершина новая
     $F(V_k) = 1$ ; // устанавливаем флаг посещаемости
     $L(V_k) = L(V) + 1$ ; // записываем номер уровня
    Добавление  $V_k$  в  $Q$ ;
  Если  $L(V_k) > L(V)$ , то: // т.е.  $V$  – родитель  $V_k$  в  $G_1'$ 
    Добавление  $V$  в  $P(V_k)$ ; // в список родителей
  Если  $V_k = V_H$ , то: // нашли нужную вершину
     $LS = L(V_H)$ ;
  Конец цикла

```

Конец цикла

5. $k = 1$, добавление индекса V_H в $M[1]$; // k – глобальная переменная
6. Запуск рекурсивной функции поиска путей $f(V_H)$;

Рекурсивная функция $f(V)$ выполняет запись в память всех кратчайших путей от вершины V до вершины V_K и имеет следующий алгоритм:

```

 $f_R(V)$  {
  Если  $(V \neq V_K)$ , то:
     $l = k$ ;  $n = \text{size}(M[k])$ ;
    Цикл по  $i$  от 1 до  $\text{size}(P(V))$ 
      Если  $i > 1$ , то:
         $k = k + 1$ ;
        Копирование  $n$  первых элементов из  $M[l]$  в  $M[k]$ ;
        Добавление индекса вершины  $P(V)[i]$  в конец  $M[k]$ ;
         $f(P(V)[i])$ ;
      Конец цикла
    }
}

```

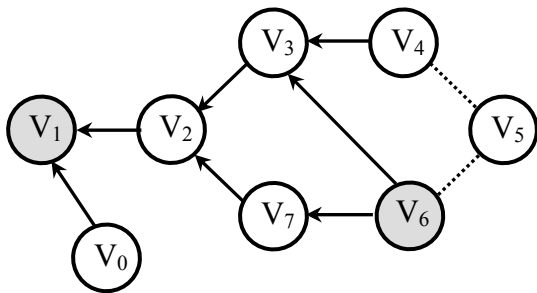


Рис. 5: Граф G_1' .

По завершению шага 6 алгоритма поиска всех кратчайших путей мы получаем k кратчайших путей из V_H в V_K , сохраненных в массиве M . В каждой строке массива M будет содержаться перечень вершин одного из кратчайших путей, начиная с V_H и заканчивая V_K .

Проиллюстрируем работу предложенного алгоритма с помощью графа G , изображенного на рисунке 3. После выполнения шагов 1-4 мы получаем в памяти граф G_1' , представленный на рисунке 5. Поскольку $L(V_5) > L(V_6)$, то построение G_1' прекращается до обработки вершины V_5 . Следовательно, она не входит в G_1' . После шага 6 получаем два кратчайших маршрута из V_6 в V_1 : $V_6 \rightarrow V_3 \rightarrow V_2 \rightarrow V_1$ и $V_6 \rightarrow V_7 \rightarrow V_2 \rightarrow V_1$. В данном случае массив M будет состоять из двух элементов (путей), каждый длиной 4 вершины.

4 СИСТЕМА НАВИГАЦИИ

Описанные алгоритмы поиска кратчайших маршрутов были протестированы нами при создании навигационной системы для космического видеотренажера, использующего систему GLView визуализации трехмерных виртуальных сцен. Данный тренажер входит в состав имитационно-тренажерного комплекса подготовки космонавтов для работы на МКС.

Начальная и целевая точки маршрута выбираются нажатием мышки на соответствующих поручнях визуализированной виртуальной модели космической станции или ее модуля. При наведении указателя мышки на поручень на экране высвечивается имя этого поручня. Если в этот момент нажать клавишу мыши, то поручень подсвечивается, что означает его выбор пользователем. Если выбрать два поручня, то между ними автоматически будут построены один или все (в зависимости от поставленной задачи) кратчайшие маршруты. Для отображения первого построенного пути между двумя поручнями осуществляется подсветка поручней, входящих в маршрут, заданным цветом. Поочередную подсветку альтернативных путей можно осуществлять нажатием клавиши Р на клавиатуре.

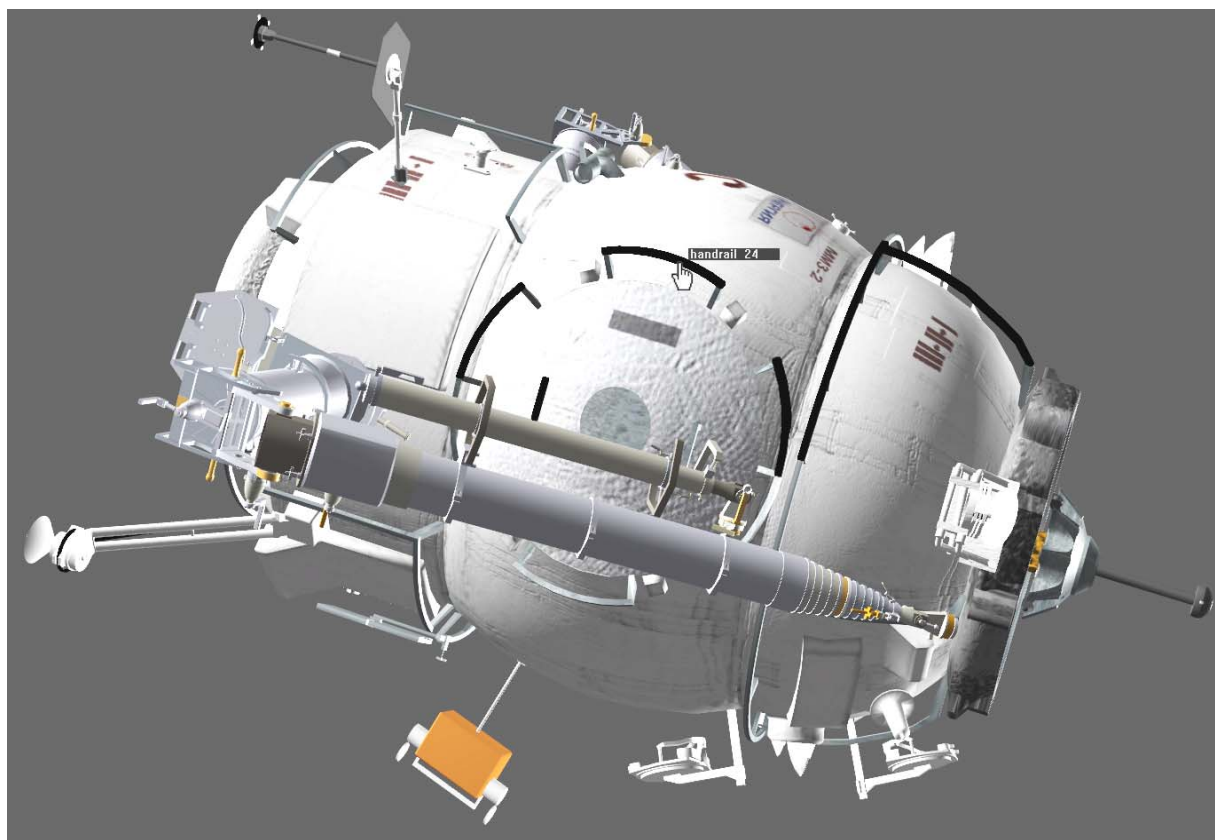


Рис. 6: Выделение маршрута на модели модуля ПИРС МКС.

На рис. 6 представлен пример работы созданной навигационной системы на модели модуля Пирс Международной космической станции (поручни, входящие в маршрут выделены черным цветом).

5 БУДУЩИЕ РАЗРАБОТКИ

Для обеспечения большей реалистичности имитируемой окружающей обстановки и более полного погружения в виртуальное пространство в тренажерных комплексах подготовки космонавтов целесообразно применять технологию виртуальной реальности. Прежде всего, эта технология включает стерео режим в шлемах виртуальной реальности. На экраны шлема осуществляется вывод изображений, синтезированных на компьютере и аналогичных тому, что видит космонавт при работе как внутри, так и на поверхности станции. Изображения в шлеме зависят от положения наблюдателя и направления его взгляда (поскольку определение последнего параметра является достаточно трудоемкой задачей, то вместо него можно учитывать ориентацию головы). Эти параметры можно определить с помощью специальных систем трекинга.



Рис. 7: Шлем и перчатки виртуальной реальности.

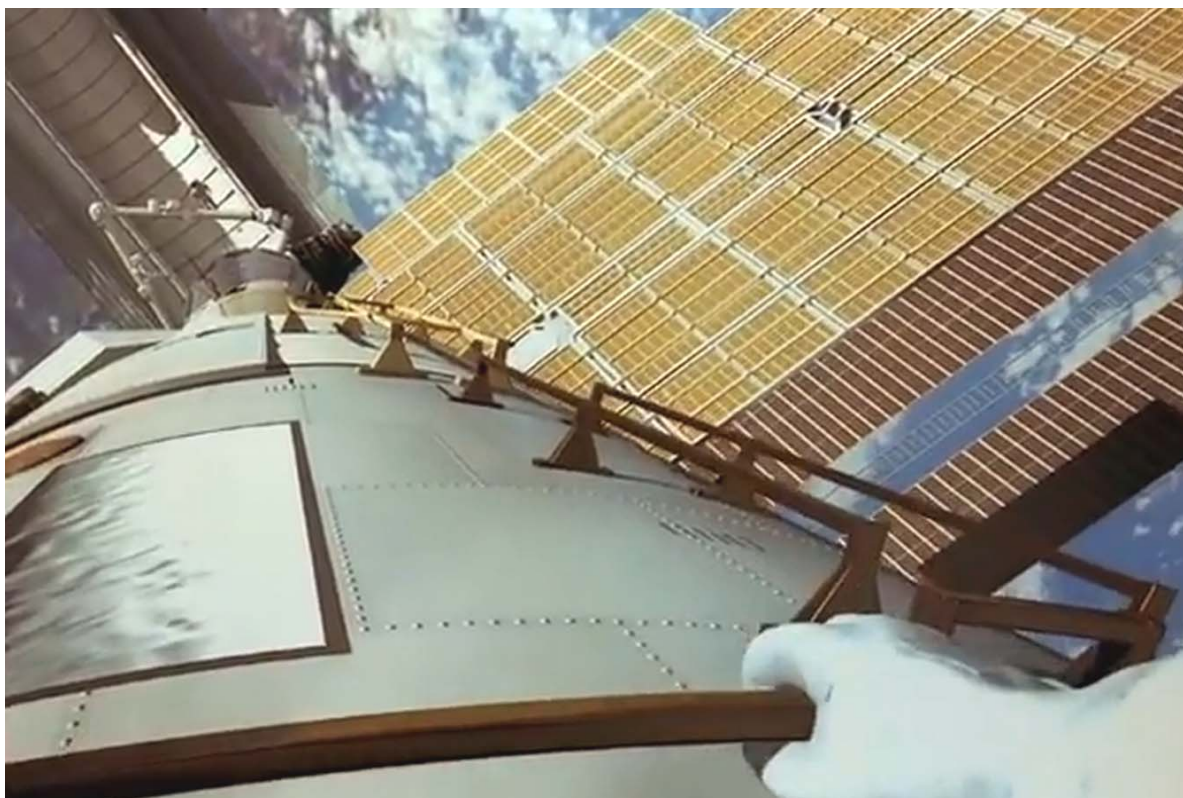


Рис. 8: Передвижение в виртуальном пространстве по внешней поверхности модели МКС.

Для обеспечения взаимодействия тренирующегося человека с виртуальной средой также возникает потребность в использовании специальных компьютерных перчаток, с помощью которых отслеживается положение и действия рук и пальцев (рис. 7), а также реализуется обратная связь (тактильные ощущения).

Одной из перспективных задач в области тренажерных комплексов подготовки космонавтов является создание системы, одновременно использующей возможности шлемов виртуальной реальности, компьютерных перчаток и навигации. При этом модели рук в виртуальном пространстве будут повторять движения реальных рук и пальцев. Производя движения руками и сгибы пальцев, тренируемый космонавт сможет брать в виртуальные руки различные объекты трехмерной сцены, захватывать поручни и перемещаться по ним (на рис. 8 показан захват поручня в виртуальном пространстве моделью правой руки). Система навигации будет подсказывать правильное направление движения, подсвечивая следующий по маршруту поручень, в зависимости от текущего положения наблюдателя.

Небольшая модификация предложенного алгоритма поиска пути между двумя поручнями позволит осуществлять выбор маршрута из всех имеющихся кратчайших путей, маршрута, проходящего через заданный поручень, и т.д. Кроме того, в обучающих целях, система навигации сможет воспроизвести движение космонавта по выбранному маршруту.

Данная работа выполняется при поддержке Программы фундаментальных исследований ОНИТ РАН, проект № 2.9.

ЛИТЕРАТУРА

- [1] Хураськин И.А. *Создание виртуального окружения в тренажерных комплексах*. Сборник докладов международной научной конференции, посвященной 80-летию со дня рождения академика В.А. Мельникова, г. Москва, 19-20 февраля 2009 г., стр. 122-125.
- [2] Михайлюк М.В., Торгашев М.А. *Система "GLView" визуализации для моделирующих комплексов и систем виртуальной реальности*. Вестник РАН, т. 11, № 2, 2011, стр. 20-28.
- [3] F. Harary, "*Graph Theory*", Addison-Wesley, 1969.
- [4] D.B. West, "*Introduction to Graph Theory*", Prentice Hall, 1996.
- [5] Zhan, F. Benjamin; Noon, Charles E. "Shortest Path Algorithms: An Evaluation Using Real Road Networks". *Transportation Science* **32** (1): 65–73 (1998).
- [6] T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2001.