УДК 004.932.2

ПРОСТРАНСТВЕННАЯ РЕКОНСТРУКЦИЯ В СИСТЕМАХ КОМПЬЮТЕРНОГО ЗРЕНИЯ НА ОСНОВЕ WEB-КАМЕР

А. В. Атанов, А. А. Крыловецкий, С. Д. Кургалин, С. И. Протасов

Воронежский государственный университет

Поступила в редакцию 25.11.2011 г.

Аннотация. Рассматривается задача реконструкции моделей объектов по дальнометрическим изображениям, полученным с web-камер, с применением метода функций уровня.
Ключевые слова: методы функций уровня, реконструкция поверхностей, машинное зрение.
Annotation. This paper deals with 3D surface reconstruction from the data obtained from measuring real-world objects with webcams. We reconstruct the 3D model of the real object based on the level-set method.
Keywords: level-set method, surface reconstruction, computer vision.

ВВЕДЕНИЕ

Создание реалистичных виртуальных моделей физических объектов – одна из основных задач компьютерного зрения. Интерес к данной проблеме объясняется большим количеством её приложений в различных областях: медицине, геологии, археологии, в задачах контроля качества, в индустрии компьютерных игр, робототехнике и т.д (см., например, [1]). Под восстановлением (или реконструкцией) здесь понимается построение физически корректной модели сцены (учитывающей геометрию, пространственное расположение, свойства поверхности и т.д.) и её представление в удобной для изменения, хранения и визуализации форме.

В процессе восстановления поверхности можно выделить три основных этапа.

1. Получение информации, т.е. измерение трёхмерных координат точек поверхности объекта с помощью, например, оптических бесконтактных измерительных устройств. Такие устройства позволяют получить так называемые дальнометрические изображения, т.е. изображения, содержащие информацию о расстоянии от камеры до конкретной точки на поверхности изучаемого объекта. Для дальнейшей реконструкции обычно требуется несколько таких изображений, полученных с различных точек.

2. Совмещение изображений: поиск преобразования, позволяющего получить точки «склейки» нескольких изображений, полученных на первом этапе для создания первой грубой трёхмерной модели объекта в одной, общей для всех системе координат.

3. Собственно реконструкция – построение модели по имеющимся точкам.

Подходы к решению задачи восстановления можно разделить на два больших класса: параметрические и непараметрические. Например, к первому классу можно отнести триангуляцию Делоне и построение диаграмм Вороного для поиска связей между неупорядоченными точками. Однако в трёхмерном случае использование подобных методов реконструкции становится довольно затруднительным и требует больших временных затрат, особенно в тех случаях, когда модель может деформироваться или содержит отверстия [2]. В последние годы большую популярность приобрели непараметрические методы, в частности, методы, использующие неявно заданные функции. Такой способ реконструкции даёт ряд преимуществ: простая структура данных, быстрота обработки, возможность легко учитывать изменения в геометрии объекта.

В данной статье решается задача реконструкции трехмерных поверхностей по данным, получаемым с простейших оптических датчиков (web-камер), непараметрическим способом – методом функций уровня.

1. ДВИЖЕНИЕ ГРАНИЦЫ

Пусть задана некоторая область Ω, как, например, на рис.1. Тогда под границей этой области будет пониматься кривая (в двумерном случае) или поверхность (в трёхмерном случае), разделяющая область на две подобласти – внешнюю, Ω^+, и внутреннюю, Ω^-.

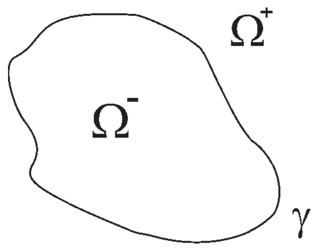

Рис. 1. Граница γ, разделяющая область Ω на подобласти

Граница может перемещаться двумя способами – либо только расширяться или сжиматься, либо и расширяться, и сжиматься одновременно. Для первого случая используется описание с помощью краевой задачи, во втором случае возникает задача Коши. Кроме того, при движении могут учитываться три компонента вектора скорости – локальный по отношению к поверхности, глобальный, зависящий от свойств поверхности, а также внешний компонент, не зависящий от вида поверхности.

Для описания движения границ, которые могут исключительно расширяться или сжиматься, используется метод Fast Marching [2,3,4], заключающийся в решении краевой задачи для уравнения эйконала:

$$|\nabla T(x)|\, F(x) = 1,$$

где T – функция прибытия, F – скорость движения границы. Функция T есть функция из R^3, описывающая границу из R^2. Если задана точка (x, y), то T будет показывать время, спустя которое граница достигнет этой точки.

В данной формулировке функция F может быть либо строго положительна, либо строго отрицательна, что соответствует либо расширению, либо сжатию границы.

Если же граница может одновременно и расширяться, и сжиматься, возможно, проходя через одну и ту же точку не один раз, то движение границы в таком случае может быть описано с помощью метода функций уровня.

2. МЕТОД ФУНКЦИЙ УРОВНЯ

Метод функций уровня был впервые рассмотрен в работе С. Ошера и Дж. Сетиана [5] и был развит в последующих работах [6, 7] как простой и универсальный способ анализа движения границы $\Gamma(t)$ некоторой замкнутой области Ω под действием поля скоростей $\vec{v}$ (в двумерном случае $\Gamma(t)$ – это кривая, в трёхмерном – поверхность). Скорость $\vec{v}$ может зависеть от геометрии границы, её положения в пространстве, а также от различных внешних сил.

Для задания границы в пространстве, в котором она расположена, вводится функция уровня $\varphi(\vec{x}, t)$, где $\vec{x}$ – координаты точки, а t – момент времени, причём положение точек, лежащих на границе, задаётся неявно уравнением

$$\varphi(\vec{x}, t) = 0. \tag{1}$$

Таким образом,

$$\varphi(\vec{x}, t) < 0, \text{ если } \vec{x} \in \Omega,$$

$$\varphi(\vec{x}, t) > 0, \text{ если } \vec{x} \in \overline{\Omega},$$

$$\varphi(\vec{x}, t) = 0, \text{ если } \vec{x} \in \partial\Omega = \Gamma(t).$$

В качестве исходных данных для реконструкции мы будем использовать множество S точек, полученных в результате сканирования реального объекта.

Первым шагом алгоритма является выбор сетки, в узлах которой и будут производиться все дальнейшие вычисления. Очевидно, что чем меньший шаг h сетки будет выбран, тем точнее будет модель восстанавливаемого объекта. Однако слишком частое разбиение может привести к значительному увеличению времени, затраченному на реконструкцию.

Следующим шагом после выбора сетки является задание начальных значений функции $\varphi(\vec{x}, t)$ в каждом узле. Для этого сначала произвольно задаётся положение границы в начальный момент времени $t = 0$, например, это может куб, окружающий все точки множества S. В нашем случае в качестве начальных значений используется расстояние d от данного узла до множества S (а именно, до ближайшей точки из S), так, что

$$\varphi(\vec{x}, t) = -d, \text{ если } \vec{x} \in \Omega_{t=0},$$

$$\varphi(\vec{x}, t) = +d, \text{ если } \vec{x} \in \overline{\Omega}_{t=0}.$$

Для получения уравнения движения кривой $\Gamma(t)$ продифференцируем уравнение (1) по переменной t. В итоге получим:

$$\frac{\partial\varphi}{\partial t} + \nabla\varphi \cdot \frac{d\vec{x}(t)}{dt} = 0. \tag{2}$$

Пусть $\vec{n}$ – внешняя нормаль к поверхности, тогда через функцию φ её можно записать следующим образом:

$$\vec{n} = \frac{\nabla\varphi}{|\nabla\varphi|}.$$

Скалярная функция F, определяющая движение в направлении внешней нормали, задаётся как

$$F = \frac{d\vec{x}(t)}{dt} \cdot \vec{n}. \qquad (3)$$

Используя запись (3), уравнение (2) можно переписать в виде:

$$\frac{\partial \varphi}{\partial t} + F \mid \nabla \varphi \mid = 0. \qquad (4)$$

В качестве функции F в нашей модели будет использоваться функция расстояния $d(x)$.

Для практического применения формулы (5) воспользуемся дискретизацией, предложенной в [1]:

$$\varphi_{ij}^{n+1} = \varphi_{ij}^{n} + \Delta t(\max(d_{ij}, 0)\nabla^{+} + \\ + \min(d_{ij}, 0)\nabla^{-}),$$

где

$$\nabla^{+} = (\max(\varphi_x^-, 0)^2 + \min(\varphi_x^+, 0)^2 + \\ + \max(\varphi_y^-, 0)^2 + \min(\varphi_y^+, 0)^2)^{1/2},$$

$$\nabla^{-} = (\min(\varphi_x^-, 0)^2 + \max(\varphi_x^+, 0)^2 + \\ + \min(\varphi_y^-, 0)^2 + \max(\varphi_y^+, 0)^2)^{1/2}.$$

$$\varphi_x^+(i,j) = \frac{\varphi(i+1,j) - \varphi(i,j)}{h},$$

$$\varphi_x^-(i,j) = \frac{\varphi(i,j) - \varphi(i-1,j)}{h},$$

$$\varphi_y^+(i,j) = \frac{\varphi(i,j+1) - \varphi(i,j)}{h},$$

$$\varphi_y^-(i,j) = \frac{\varphi(i,j) - \varphi(i,j-1)}{h}.$$

В результате на каждой итерации поверхность, которая была выбрана в качестве исходного приближения, будет изменяться и двигаться по направлению к исходному множеству точек S, формируя в конечном итоге модель реконструируемого объекта.

3. РЕКОНСТРУКЦИЯ

Для формирования исходных данных был получен набор дальнометрических изображений с web-камер. Особенностью таких изображений является то, что каждый пиксел содержит величину расстояния от опорной точки до соответствующей точки на объекте съёмки (подробнее о визуализации по дальнометрическим изображениям см.[8]).

Реализация метода функций уровня, описанного в п.2, была выполнена в среде Matlab R2010a. В результате работы программы были решены соответствующие уравнения и получен набор точек, по которому реконструирована исследуемая поверхность.

Для вычислений использовался компьютер с процессором Intel Pentium IV. Время, затраченное на получение итоговой модели рис. 2(г), составило около 13 часов.

ЗАКЛЮЧЕНИЕ

Как правило, метод функций уровня используется для реконструкции поверхностей по точкам, полученным с высокоточных и дорогостоящих сканеров. В данной работе была рассмотрена возможность применения данного метода при построении трёхмерных моделей объектов с использованием обычных веб-камер, использование которых существенно снижает стоимость любого устройства компьютерного зрения.

Было установлено, что метод функций уровня позволяет получать удовлетворительные результаты реконструкции даже при использовании исходных данных низкого качества. Однако для того, чтобы итоговая модель не имела значительных искажений, перед реконструкцией исходные данные подверглись дополнительной обработке – удаление шума, неизбежно возникающего при сканировании, отбрасывание значительно удалённых от объекта изолированных точек, а также масштабирование объекта. Кроме того, для подсчёта начального значения $d(x)$ был использован метод Fast Sweeping [9], позволяющий быстро и точно получить данные о расстоянии от каждой точки сетка до множества исходных данных S.

Построение модели в системах компьютерного зрения, как правило, должно выполняться достаточно быстро, однако в ходе проверки работы алгоритма, реализующего метод функций уровня, было обнаружено, что время, затрачиваемое на реконструкцию моделей объектов, имеющих даже малый размер, оказывается неприемлемо большим.

Таким образом, для практического применения метода функций уровня в реальных приложениях необходимо сократить время работы алгоритма, на каждом шаге которого приходится решать уравнение в частных производных для каждого узла сетки. В ряде работ (см, например, [10]) были предложены модификации метода функций уровня, в которых все

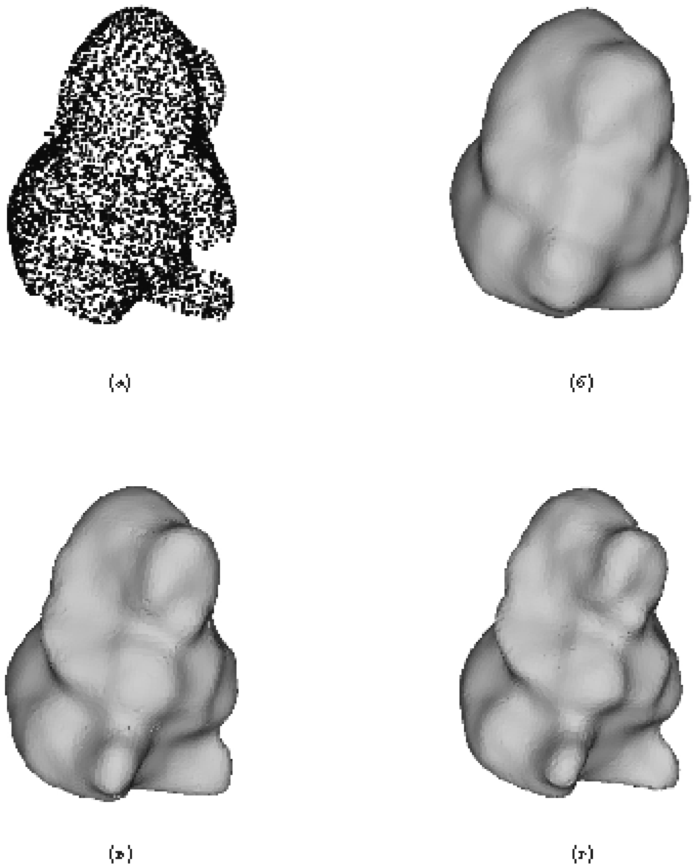

Рис. 2. (а) – множество точек, полученных в результате сканирования объекта с применением дальнометрических изображений, (б), (в) ,(г) – этапы реконструкции объекта с применением метода функций уровня на 2200-й, 3200-й и 3400-й итерациях соответственно

вычисления производятся лишь в некоторой окрестности точек исходного множества S, что существенно ускоряет работу алгоритма. Однако такой подход не всегда применим в реальных задачах. Время, затрачиваемое на реконструкцию, также может быть сокращено за счёт применения параллельных вычислений, например, таким образом можно существенно ускорить расчеты с использованием разностных схем, описанных в п.2. Поиск возможных модификаций, позволяющих уменьшить объем вычислений и в тоже время универсальных, позволит приблизиться к созданию систем трехмерной реконструкции реального времени.

СПИСОК ЛИТЕРАТУРЫ

1. *Paragios N.* Handbook of Mathematical Models in Computer Vision / N.Paragios, Y.Chen, O.D.Faugeras. – Springer. – 2006. – 639 p.

2. *Sethian J.A.* Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. / J.A. Sethian. – University of California, Berkeley. – 1999. – 404 p.

3. *Sethian J.A.* Evolution, Implementation, and Application of Level Set and Fast Marching Methods for Advancing Fronts. / J.A. Sethian. // J. Comp. Phys., vol. 169. – 2001. – pp. 503–555.

4. *Sethian J.A.* Fast Marching Methods. / J A. Sethian. // SIAM Review, Vol. 41, No. 2. – 1999. – pp. 199–235.

5. *Osher S.* Fronts Propagating with Curvature Dependent speed: Algorithms Based on Hamilton-Jacobi Formulation. / S. Osher, J.A. Sethian // J. Comp. Phys., 79. – 1988. – pp. 12–49.

6. *Zhao H.K.* Implicit and Non-parametric Shape Reconstruction from Unorganized Points Using Variational Level Set Method. / H.K. Zhao, S. Osher, B. Merriman, M. Kang // Computer Vision and Image Understanding. Vol. 80. – 2000. – pp. 295–319.

7. *Osher S.* Level Set Methods and Dynamic Implicit Surfaces. / S. Osher, R.P. Fedkiw. – Springer. – 2002. – 296 p.

8. *Карапиш Г.А.* Построение изоповерхности по совмещенным дальнометрическим изображениям. / Г.А. Карапиш, А.А. Крыловецкий // Технологии Microsoft в теории и практике программирования: тр. конф. студентов, аспирантов и молодых ученых. – М.: МАИ. – 2008.

9. *Zhao H.* A fast sweeping method for eikonal equations / H. Zhao // Mathematics of Computation, Vol. 74. – 2005. – pp. 603–627.

10. *Peng D.* A PDE-Based Fast Local Level Set Method. / D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang // J. Comp. Phys., 155. – 1999. – pp.410–438.

Атанов Артем Викторович – аспирант кафедры цифровых технологий Воронежского государственного университета, e-mail: aatanov@mail.ru.

Крыловецкий Александр Абрамович – к.ф.-м.н., доцент каф. цифровых технологий Воронежского государственного университета, e-mail: e-mail: aakryl@cs.vsu.ru

Кургалин Сергей Дмитриевич – д.ф.-м.н., заведующий кафедрой цифровых технологий Воронежского государственного университета, e-mail: kurgalin@bk.ru

Протасов Станислав Игоревич – аспирант кафедры цифровых технологий Воронежского государственного университета, e-mail: stanislav.protasov@gmail.com

Atanov A.V. – Postgraduate student of Department of Digital Technologies, Voronezh State University, e-mail: aatanov@mail.ru.

Krylovetsky A.A. – Ph.D. in Physics and Mathematics, Associate Professor of Department of Digital Technologies, Voronezh State University, e-mail: aakryl@cs.vsu.ru

Kurgalin S.D. – Doctor of Physics and Mathematics, Head of the Department of Digital Technologies, Voronezh State University, e-mail: kurgalin@bk.ru.

Protasov S. I. – Postgraduate student of Department of Digital Technologies, Voronezh State University, e-mail: stanislav.protasov@gmail.com.