05.13.18

И.С. Стародубцев

Уральский федеральный университет им. Б.Н. Ельцина, институт математики и компьютерных наук, кафедра информатики и процессов управления, Екатеринбург, StarodubtsevIS@ya.ru

Инициализируемый захват движения и слежение на основе карт глубин для жестового интерфейса

В статье представлен метод захвата движения и трекинга точки интереса на основе карт глубин с помощью инициирующих жестов. Описанный метод является частью комплексного жестового интерфейса.

Ключевые слова: *карта глубин, жестовый интерфейс, движение, трекинг*.

Введение

Описанный ниже метод является одним из основных методов захвата движения для комплексного жестового интерфейса. В основе данного метода лежит идея «инициализации» точки интереса с помощью специального жеста.

Предполагается, что в видимой сцене присутствует оператор, находящийся ближе к датчику, чем видимый фон. Так же предполагается, что сцена не подвержена значительным резким изменениям.

Метод построен на анализе последовательности кадров, содержащих карты глубин сцены. Карту глубин будем представлять, как функцию вида

$$0<f_t(x,y)\le\infty \quad , \qquad (1)$$

где $f_t(x,y)$ – значение глубины сцены в точке, соответствующей точке (x,y) кадра; t – временная отметка кадра; x, y – пространственные координаты в системе координат кадра. Если для некоторой точки (x_0,y_0) по каким-либо причинам нет возможности вычислить глубину, то полагаем $f_t(x_0,y_0)=\infty$.

Данные такого типа возможно получать с различных датчиков, например со сканирующих лазерных дальномеров или с устройств на основе PSDK (Asus XTion, MS Kinect).

Разработкой подобных алгоритмов захвата движения занимается большое количество как индивидуальных, так и корпоративных разработчиков. В качестве примера привести проекты MS Kinect SDK и OpenNI+NITE, также использующие карты глубин для захвата движения и слежения. Однако большинство подобных разработок не могут быть верифицированы, из-за отсутствия оценок точности, сложности и быстродействия или ограниченного доступа к ним. Другой проблемой такого рода разработок является, как правило, жёсткие ограничения на условия использования или привязка к определённым типам датчиков.

Инициируемый метод слежения

Метод состоит из последовательного выполнения этапов начальной инициализации точки интереса и последующего трекинга этой точки.

Детектор движения

На этапе инициализации производится первоначальный выбор точки интереса. В основе лежит идея инициализации пользователем слежения за определённым объектом с помощью ключевого жеста. В качестве таких ключевых жестов используются три: «горизонтальная волна», «вертикальная волна» и «нажатие».

Для распознавания ключевых жестов необходимо выделить движение в последовательности кадров, выделить движущийся объект и начать слежение за ним. Один из простейших подходов к обнаружению изменений в последовательности кадров, произошедших между кадрами $f_{t_i}(x,y)$ и $f_{t_j}(x,y)$, полученными в моменты t_i и t_j состоит в поэлементном сравнении этих кадров. Однако метод простого вычитания

$$diff\left[f_{t_i},f_{t_j}\right](x,y)=\begin{cases}1, \left|f_{t_i}(x,y)-f_{t_j}(x,y)\right|>T_{diff}\\ 0, \left|f_{t_i}(x,y)-f_{t_j}(x,y)\right|\le T_{diff}\end{cases} \quad (2)$$

даёт неудовлетворительные результаты, отчасти из-за шума, отчасти из-за, как ни странно, оператора, мелкие движения которого могут быть ошибочно интерпретированы как жест.

Для борьбы с этими проблемами были выбраны «повторные» жесты,

которые легко воспроизвести, но относительно сложно сделать это случайно. Для регистрации этих жестов используются *накопительные разностные буферы* [1, 2]. Идея состоит в том, чтобы игнорировать изменения, встречающиеся однократно и изредка — т. к. скорее всего такие изменения являются шумом или «случайными» движениями оператора [3,4].

Рассмотрим последовательность кадров $\{f_{t_i}\}_{i=1}^{n}$. Обозначим для простоты записи $f_{t_i}=f(x,y,i)$. Пусть $R(x,y)=f(x,y,n)$. Тогда для любого $k>1$ значения элементов буфера в каждой точке (x,y) определяются следующим образом

$$A_k=\begin{cases} A_{k-1}(x,y)+1 & |R(x,y)-f(x,y,n-k)|>T_{buf}, \\ A_{k-1}(x,y) & |R(x,y)-f(x,y,n-k)|\le T_{buf}, \end{cases}$$

$$P_k=\begin{cases} P_{k-1}(x,y)+1 & (R(x,y)-f(x,y,n-k))>T_{buf}, \\ P_{k-1}(x,y) & (R(x,y)-f(x,y,n-k))\le T_{buf}, \end{cases} \qquad (3)$$

$$N_k=\begin{cases} N_{k-1}(x,y)+1 & (R(x,y)-f(x,y,n-k))<-T_{buf}, \\ N_{k-1}(x,y) & (R(x,y)-f(x,y,n-k))\ge -T_{buf}, \end{cases}$$

где A_k, P_k и N_k – абсолютный, позитивный и негативный накопительные разностные буферы, соответственно, $A_1 \equiv 0$, $P_1 \equiv 0$, $N_1 \equiv 0$; $T_{buf}(x,y) = const$ – пороговая функция [5].

Поскольку инициирующий жест является повторяющимся и сцена не подвержена значительным изменениям во время съёмки, присутствуют большие значения буферов в области жеста на фоне незначительных значений в остальной области кадра.

Также для дальнейшего анализа нам понадобится выделить сам объект интереса в области, где происходило движение. Мы можем сделать это, используя предположение, что объект интереса находится в сцене ближе к датчику, нежели объекты фона. Для этого проведём сегментацию множества точек $A^{\delta}=\{(x,y): A_k(x,y)>\delta\}$, где $\delta \ge 0$ с помощью пороговой обработки по глубине с порогом $T(x,y)$. Значение пороговой функции $T(x,y)$ может быть выбрано из различных соображений [6, 7, 8, 9].

После сегментации получены две группы точек:

$$G_1=\{(x,y):(x,y)\in A^{\delta},f_t(x,y)\geq T(x,y)\},$$
$$G_2=\{(x,y):(x,y)\in A^{\delta},f_t(x,y)<T(x,y)\}. \quad (4)$$

Группа G_1 будет соответствовать точкам фона, а группа G_2 будет соответствовать точкам объекта интереса.

Итак, пусть I_t — искомая точка интереса. Обозначим её координаты (x_I,y_I) . Тогда они могут быть вычислены с использованием следующих формул:

$$x_I=a\frac{\sum_x\sum_y P_k(x,y)S_P(x,y)x}{\sum_x\sum_y P_k(x,y)S_P(x,y)}+(1-a)\frac{\sum_x\sum_y N_k(x,y)S_N(x,y)x}{\sum_x\sum_y N_k(x,y)S_N(x,y)},$$
$$y_I=a\frac{\sum_x\sum_y P_k(x,y)S_P(x,y)y}{\sum_x\sum_y P_k(x,y)S_P(x,y)}+(1-a)\frac{\sum_x\sum_y N_k(x,y)S_N(x,y)y}{\sum_x\sum_y N_k(x,y)S_N(x,y)}, \quad (5)$$

где $a\in[0,1]$ — параметр, отвечающий за вес опорного кадра; суммирование ведётся по пространственным координатам области кадра; S_P, S_N — весовые функции, позволяющие более тонко манипулировать областью и параметрами поиска движения. В частности, с их помощью можно задавать маски областей, в которых движение рассматриваться не будет. Имеет смысл при построении использовать множества G_1 и G_2, увеличивая вес точек, соответствующих объекту интереса и уменьшая вес точек, соответствующих фону.

Таким образом, в целом процедура инициализации точки интереса при инициируемом подходе выглядит следующим образом:

При поступлении нового *t*-го кадра $f_t(x,y)$, предполагаем, что в этом кадре завершается жест инициализации, производимый в течении *n* кадров. В качестве опорного кадра выбирается текущий кадр, $R(x,y)=f_t(x,y)$. Далее по *n* кадрам пересчитываются накопительные разностные буферы *A*, *N* и *P*, и производится попытка инициализации точки интереса по формулам (3) и если полученная точка *I* с координатами (x_I,y_I) удовлетворяет условию

$$(x_I, y_I) \in G_2, \tag{6}$$

то за точкой начинается слежение. В противном случае инициализация считается не успешной и ожидается следующий кадр.

Трекинг

После успешной инициализации точки интереса её координаты передаются модулю слежения, в задачи которого входит сопровождение точки в кадре, обработка ситуации с потерей, кратковременной и долговременной и предоставление данных о наличии и статусе точки интереса (*tracked/untraced*) и трёхмерные координаты $(x, y, f_t(x, y))$, в случае *tracked*.

Для описания алгоритма слежения за точкой используем определение *пространственной окрестности точки* $(x_{0,} y_0)$:

$$O_r(x_{0,} y_0) = \left\{(x, y) : dist\left[(x, y, f(x, y)), (x_{0,} y_{0,} f(x_{0,} y_0))\right] \le r\right\}, \tag{7}$$

где в качестве функции расстояния $dist\left[(x_{1,} y_{1,} z_1), (x_{2,} y_{2,} z_2)\right]$ можно выбрать различные метрики, например, метрику L_∞ [10], т. к. она требует меньше вычислительных ресурсов.

Для слежения используется абсолютный накопительный разностный буфер (3), полученный по двум кадрам: текущему $f_t(x, y)$ и предыдущему $f_{t-1}(x, y)$.

Для удобства записи, обозначим точку интереса на кадре $f_t(x, y)$ как (x_I^t, y_I^t). Итак для нахождения точки интереса (x_I^t, y_I^t) на текущем кадре в предположении, что известна (x_I^{t-1}, y_I^{t-1}), необходимо определить область поиска. Для этого проведём количественную оценку движения в *r* окрестности точки (x_I^{t-1}, y_I^{t-1}) и вычислим

$$\begin{aligned} x_m &= \frac{\sum_{(x,y) \in O_r(x_I^{t-1}, y_I^{t-1})} A_2(x, y)\, x}{\sum_{(x,y) \in O_r(x_I^{t-1}, y_I^{t-1})} A_2(x, y)}, \\ y_m &= \frac{\sum_{(x,y) \in O_r(x_I^{t-1}, y_I^{t-1})} A_2(x, y)\, y}{\sum_{(x,y) \in O_r(x_I^{t-1}, y_I^{t-1})} A_2(x, y)}. \end{aligned} \tag{8}$$

Уточнение центра региона поиска с учётом истории:

$$x_c = \alpha\, x_m + (1-\alpha)\, x_I^{t-1},$$
$$y_c = \alpha\, y_m + (1-\alpha)\, y_I^{t-1}. \qquad (9)$$

Произвольно изменяя параметры $\alpha \in [0,1]$, $r \in ¿$ можно регулировать влияние движения на поиск точки.

Далее происходит поиск нового положения точки интереса в $\rho \in ¿$ окрестности точки (x_c, y_c). Пусть $M = \max_{(x,y)\in O_\rho(x_c,y_c)} \{f(x,y)\}$.Тогда

$$x_I^t = \frac{\sum_{(x,y)\in O_\rho(x_c,y_c)} S_\rho(x,y)(M - f(x,y))x}{\sum_{(x,y)\in O_\rho(x_c,y_c)} S_\rho(x,y)(M - f(x,y))},$$
$$y_I^t = \frac{\sum_{(x,y)\in O_\rho(x_c,y_c)} S_\rho(x,y)(M - f(x,y))y}{\sum_{(x,y)\in O_\rho(x_c,y_c)} S_\rho(x,y)(M - f(x,y))}, \qquad (10)$$

где $S_\rho(x,y)$ - весовая функция.

Если выполнено $(x_I^t, y_I^t) \in G_2$, то полученные координаты будут новыми координатами точки интереса в кадре $f_t(x,y)$, иначе считается, что точка потеряна.

Заключение

Приведённый выше алгоритм имеет оценку по сложности и по объёму потребляемой памяти порядка *O(N)*, где N — количество точек в кадре, что позволяет использовать его в приложениях реального времени и быстрого отклика.

Реализация метода используется в прототипе жестового интерфейса для медицинских задач [11], и имеет положительные оценки от специалистов.

Список литературы

[1] *Chris Stauffer, W. Eric L. Grimson* Learning Patterns of Activity Using Real-Time Tracking // IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2000. P. 747-757

[2] *Omar Javed, Khurram Shafique, Mubarak Shah* A Hierarchical Approach to Robust Background Subtraction using Color and Gradient Information // Proceedings of the Workshop on Motion and Video Computing, IEEE Computer Society Washington, DC, USA, 2002. P. 22

[3] *Dirk Focken, R. Stiefelhagen* Towards Vision-based 3-D People Tracking in a Smart Room // ICMI '02 Proceedings of the 4th IEEE International Conference on Multimodal Interfaces, IEEE Computer Society Washington, DC, USA 2002. P. 400

[4] *Alexandre R.J. Francois and Re R. J. François and Gerard G. Medioni* A Modular Software Architecture for Real-Time Video Processing // In IEEE International Workshop on Computer Vision Systems, Springer, 2001. P. 35-49.

[5] *Гонсалес Р., Вудс Р*. Цифровая обработка изображений. — М.: Техносфера, 2005, 2006. -1072 с.

[6] *Потапов А. А., Пахомов А. А., Никитин С. А., Гуляев Ю. В*. Новейшие методы обработки изображений. — М.: Физматлит, 2008.

[7] *Степаненко О. С.*, Сканеры и сканирование. Краткое руководство. — М.: Диалектика, 2005.

[8] *Иванов Д. В., Хропов А. А., Кузьмин Е. П., Карпов А. С., Лемпицкий В. С.* Алгоритмические основы растровой графики, 2007. Учебное пособие.

[9] *Дьяконов В. П.* MATLAB 6.5 SP1/7/7 SP1/ Работа с изображениями и видеопотоками. — М.: СОЛОН-Пресс, 2010.

[10] *Колмогоров А.Н., Фомин С.В.* Элементы теории функций и функционального анализа. — изд. четвёртое, переработанное. — М.: Наука, 1976. — 544 с.

[11] Averbukh V., Starodubtsev I., Tobolin D., The Gesture Interface For Control Of Angiographic Systems // Современные компьютерные и информационные технологии: сборник трудов международной научной Российско-Корейской конференции. Екатеринбург: УрФУ, 2012, с.97 – 107.

I. S. Starodubtsev

Depth map based initialized motion capture and tracing for gesture interface

Keywords: *depth scene map, motion capture, gesture interface, tracking*

The paper presents a method of depth based motion capture and tracking points

of interest using initiating gestures. The described method is part of a complex gestures interface.