

Санкт-Петербургский Государственный Университет
Математико-Механический Факультет

Кафедра системного программирования

НЕСТАЦИОНАРНАЯ ОПТИМИЗАЦИЯ
РАНДОМИЗИРОВАННЫМИ АЛГОРИТМАМИ ПРИ
ЛИНЕЙНОМ ПРЕДСКАЗАНИИ ЭКСТРЕМУМА В
ЗАДАЧАХ КОМПЬЮТЕРНОГО ЗРЕНИЯ

Дипломная работа студента 545 группы

Кривоконь Дмитрия Сергеевича

Научный руководитель к.ф.-м.н., Вахитов А. Т.
/ подпись /

Рецензент Гуревич Л. С.
/ подпись /

“Допустить к защите” д.ф.-м.н., проф. Терехов А.Н.
заведующий кафедрой, / подпись /

Санкт-Петербург
2011

**Saint-Petersburg State University
Mathematics and Mechanics Faculty**

Software Engineering Chair

**NONSTATIONARY OPTIMISATION USING
RANDOMISED ALGORITHMS WITH LINEAR
EXTREMUM PREDICTION IN COMPUTER VISION
PROBLEMS**

Graduate paper by

Krivokon Dmitry
545 group

Supervisor PhD, A. T. Vakhitov

Reviewer L. S. Gurevich

“Approved by” PhD, Professor A. N. Terekhov

Head of Department

Saint-Petersburg
2011

Оглавление

ВВЕДЕНИЕ	4
ПОСТАНОВКА ЗАДАЧИ	5
ГЛАВА I. ОТСЛЕЖИВАНИЕ ПОЛОЖЕНИЯ ОБЪЕКТОВ СТЕРЕОКАМЕРОЙ	6
Задачи компьютерного зрения	6
<i>Детектирование и отслеживание объектов</i>	6
<i>Отслеживание объекта при помощи двух камер</i>	10
Фильтрация динамических систем.....	11
Стохастическая аппроксимация	13
Модель наблюдения за объектом на основе двух камер	15
<i>Математическая модель представления камер</i>	15
<i>Определение положения точки на основе двух камер</i>	17
ГЛАВА II. АЛГОРИТМ ФИЛЬТРАЦИИ ДИНАМИЧЕСКИХ СИСТЕМ С ЛИНЕЙНЫМ ПРОЦЕССОМ ИЗМЕНЕНИЯ СОСТОЯНИЯ	20
Постановка задачи фильтрации.....	20
Модифицированный алгоритм нестационарной стохастической оптимизации.....	20
Отслеживание объекта при помощи двух камер	21
ГЛАВА III. ПРАКТИЧЕСКИЕ РЕЗУЛЬТАТЫ	23
Тестирование алгоритма.....	23
Сравнение с фильтром Калмана	29
ЗАКЛЮЧЕНИЕ	32
СПИСОК ЛИТЕРАТУРЫ	33
ПРИЛОЖЕНИЕ I. ФИЛЬТР КАЛМАНА	36

Введение

Во многих системах компьютерного зрения возникает необходимость отслеживания реального положения объекта с течением времени. Чаще всего решение этой задачи осложняется присутствием различных помех. Изображения, получаемые с видеокамер, обычно не обладают высоким качеством, поэтому алгоритмы, детектирующие объекты на изображении, будут давать сильно зашумленные результаты. При отсутствии точного позиционирования объекта на изображении сложно восстановить его настоящее положение с какой-либо приемлемой точностью. Так, различные методы триангуляции, которые используются для вычисления положения объекта по его изображениям с разных ракурсов, работают крайне нестабильно даже при незначительном уровне помех [32]. Если же известен характер движения точки, то можно производить более точные оценки ее положения при помощи алгоритмов фильтрации [28,29]. Алгоритмы фильтрации рассматривают некоторую динамическую систему и ее состояние, которое известным образом меняется с течением времени. На основе зашумленных наблюдений (характер которых также известен) алгоритмы фильтрации строят оценки, которые наилучшим образом удовлетворяют произведенным наблюдениям. Движущийся объект можно рассматривать как такую систему с состоянием, включающим в себя его положение и, к примеру, скорость, а в качестве наблюдений использовать положение изображения объекта на плоскостях камер.

Задачу фильтрации можно сформулировать как задачу поиска минимума некоторого изменяющегося во времени функционала по его зашумленным измерениям. В оптимизации для решения такого рода задач существуют различные методы стохастической аппроксимации [30]. Авторы [31] предложили метод, позволяющий отслеживать минимум нестационарного функционала $F(x_n, w_n, n)$ по зашумленным наблюдениям $y_n = F(x_n, w_n, n) + v_n$ на основе алгоритма с пробными одновременными возмущениями. Предложенный алгоритм никак не учитывает известный характер движения точки минимума

функционала, что ограничивает его применимость для решения задач фильтрации динамических систем.

Постановка задачи

Перед данной работой ставились следующие задачи: разработка и реализация алгоритма отслеживания объекта с линейным характером движения при помощи двух камер на основе метода нестационарной стохастической оптимизации; реализация модели, симулирующей работу двух камер и движения точки в области их видимости; тестирование алгоритма на модельных данных при различных вариантах помех и траекторий движения точки; сравнение результатов работы алгоритма с существующим методом нестационарной стохастической аппроксимации и нелинейным фильтром Калмана. Хотя известны успешные результаты применения в задача отслеживания объекта расширенного фильтра Калмана [15,16,17,18], при его настройке требуется большое число параметров, зачастую неизвестных, и при высоких размерностях вычисление оценок требует значительных вычислительных ресурсов для обращения матриц. Цель дипломной работы - продемонстрировать, что новый алгоритм в аналогичных условиях требует меньших знаний о системе и меньшего объема ресурсов для вычислений

Глава I. Отслеживание положения объектов стереокамерой

Задачи компьютерного зрения

Детектирование и отслеживание объектов

За последние несколько лет количество приложений, в которых возникают задачи анализа изображений и видео, чрезвычайно возросло. Производительность современных вычислительных устройств позволяет создавать программное обеспечение, решающее задачи распознавания лиц, отслеживания объектов, трехмерной реконструкции и т.п., на обычном персональном компьютере. Вследствие этого приложения становятся более доступными и, соответственно, на них повышается спрос. Все это приводит к необходимости разработки новых подходов и алгоритмов, решающих те или иные задачи компьютерного зрения. Среди всех задач компьютерного зрения выделяются задачи детектирования, классификации и отслеживания объектов. Необходимость выделения на изображении отдельных частей, которые несут некую смысловую нагрузку (человек, автомобиль, движущийся объект и т.п.), возникает во множестве приложений.

Так, в охранных системах крайне важной задачей является определение движущихся объектов. Камеры в таких системах видеонаблюдения чаще всего находятся в статическом положении, поэтому имеется возможность, анализируя изменение изображения с течением времени, выделять неизменяющиеся области или фон. Получив некоторую модель фона, и проанализировав ее разность с текущим кадром, можно выделить области, в которых предполагается наличие движения. Простейшими методами моделирование фона являются среднее [1,2] и скользящее среднее. Данные алгоритмы вычисляют значение пикселя фона усреднением его значений (или усреднением с весами) по некоторому набору кадров:
$$B_N(p) = \frac{1}{N} \sum_{i=1}^N F_i(p).$$
 Области с движением в таких подходах

определяются пороговым значением для модуля разности фона с текущим кадром: $Motion(p) = |B_{N-1}(p) - F_N(p)| > threshold$. Очевидной проблемой данных методов является отсутствие какой-либо адаптивной процедуры выбора порогового значения. При изменении освещения или погодных условий стабильность систем основанных на таких алгоритмах будет значительно падать. Моделирование фона в виде одного [3] или нескольких [4] гауссовских распределений решает проблемы выбора порогового значения путем использования в качестве него дисперсии пикселя умноженной на некоторый коэффициент. Каждый пиксель фона представляется двумя параметрами распределения: B^{mean} – математическое ожидание, B^{var} – дисперсия. Обновление параметров распределения производится при помощи метода экспоненциального скользящего среднего:

$$B_N^{mean} = \alpha * F_N + (1 - \alpha) * B_{N-1}^{mean}$$

$$B_N^{var} = \alpha * (F_N - B_{N-1}^{mean})^2 + (1 - \alpha) * B_{N-1}^{var},$$

где $\alpha \in [0,1]$ задает скорость обучения модели. Выбрав некоторый коэффициент k , движение детектируется соответствующим образом:

$$Motion(p) = |B_{N-1}^{mean}(p) - F_N(p)| > k * B_{N-1}^{var}(p).$$

Кроме моделирования параметров распределения существует методы позволяющие искать приближение функции плотности вероятности пикселей фона [5]. Плотность вероятности ищется путем построения гистограммы значений пикселя и сглаживании ее при помощи специальной функции – ядра. На основе построенной функции присутствие движения определяется условием $PDF(p) < threshold, threshold \in [0,1]$.

В области классификации и распознавания объектов большинство подходов, так или иначе, основано на использовании некоторой параметризации

объектов. Выбор параметров необходимо производить таким образом, чтобы их значения сохранялись для одного и того же объекта снятого в разных условиях (ракурс, шум, освещение и т.п.).

Широко распространено представление объектов в виде набора специальных точек на его изображении – особенностей. Методы SIFT [6] и SURF [7] представляют особенности в виде точки и дескриптора, вычисленного по некоторой ее окрестности. Выбранные особенности отличаются высокой степенью воспроизводимости при различных геометрических искажениях изображения. Эти подходы широко применяются в задачах распознавания объектов [6][8].

При решении задач классификации и детектирования объектов часто используются так называемые Haar-like features [9]. Данный алгоритм вокруг произвольной точки изображения рассматривает некоторую прямоугольную окрестность фиксированного размера, эта область разбивается на определенное число прямоугольников, в каждом из которых подсчитывается сумма значений интенсивности пикселей. После этого в качестве дескриптора особенности в выбранной точке используется набор разностей сумм между смежными прямоугольниками. Основным преимуществом такого рода особенностей является то, что их дескриптор можно вычислять за константное время для произвольного размера окрестности. Такая производительность достигается за счет использования специального массива:

$$A(x, y) = \sum_{x'=0}^x \sum_{y'=0}^y \text{Img}(x', y').$$

При помощи него для произвольного прямоугольника на изображении сумма значений интенсивности пикселей в нем считается следующим образом:

$$\text{Sum}(\text{Rect}(x, y, x', y')) = A(x', y') - A(x, y) - A(x', y) + A(x, y').$$

Высокая скорость вычисления Haar-like features позволяет строить на основе них эффективные алгоритмы классификации и поиска объектов определенного типа. Используя методы машинного обучения, такие как AdaBoost [9,12], алгоритмы на основе Haar-like features успешно применяются для решения задач детектирования лиц [11] и автомобилей [12].

При анализе видеоданных часто возникает задача отслеживания объектов с течением времени. Множество приложений основаны на знании траекторий объектов и различных их динамических характеристик. В качестве простейшего решения данных задач можно использовать несколько модифицированные алгоритмы детектирования движущихся объектов [19,20]. Имея списки движущихся объектов на паре смежных кадров, можно производить сопоставление объектов между собой на основе некоторого признака (расстояние между объектами, разница между некоторыми окрестностями вокруг положения объекта, функции расстояния в терминах SIFT и SURF особенностей). Такой подход будет применим лишь в случае небольшого числа объектов. При увеличении количества движущихся объектов задача их сопоставления между кадрами становится трудно разрешимой. В качестве альтернативы данному подходу существуют методы, основанные на локальном поиске. Данные алгоритмы по приближению положения объекта с предыдущего кадра оптимизируют некоторую функцию качества по определенному набору параметров задающих положение объекта. К примеру, в алгоритме Лукаса-Канады [21] объект представляет собой некоторое изображение (образец), а в качестве оптимизируемой функции используется: $\sum_x [I(W(x,p)) - T(x)]^2$, где T – изображение объекта, I – изображение в котором ведется поиск, W - некоторое преобразование координат T в координаты I , параметры которого являются аргументами оптимизации.

Изображение объекта можно рассматривать как выборку некоторой случайной величины в определенном пространстве свойств (цвет, частоты

преобразования Фурье). Используя методы оценивания функции плотности вероятности случайных величин (Kernel-density estimation [22,23]), задачу отслеживания объектов можно свести к задаче минимизации расстояния, заданного на функциях плотности вероятности (Bhattacharyya distance [24]). Авторы [25] предложили общую схему для реализации методов отслеживания объектов, основанных на данном подходе. Такие алгоритмы хорошо работают в условиях большого числа объектов и при частичном перекрытии их изображений, но при наличии однородных по цвету или текстуре объектов они будут давать некорректные результаты.

Отслеживание объекта при помощи двух камер

Использование стереосистем камер для восстановления реального положения движущегося объекта широко распространено в робототехнике [13] и автомобильной индустрии [14]. На основе известных координат проекций объекта в плоскостях камер его настоящее положение в пространстве можно определять при помощи различных методов триангуляции. Обычно позиционирование объекта на изображениях осложнено присутствием разнообразных помех, что приводит к неудовлетворительным оценкам алгоритмов, восстанавливающих его трехмерные координаты. При известном характере движения объекта получаемые оценки можно сильно улучшить, используя различные методы фильтрации динамических систем. Так, авторы [15] представили подход к использованию фильтра Калмана в задаче отслеживания объекта двумя камерами с известными параметрами. В данной статье рассматривалась линейная динамическая система с состоянием, включающим в себя положение и скорость объекта, по которой производятся наблюдения в виде восстановленных методом триангуляции трехмерных координат объекта. Для решения задачи отслеживания объекта использовался линейный фильтр Калмана, примененный к задаче фильтрации состояния построенной системы. Основным недостатком такого подхода является то, что в качестве наблюдения рассматриваются восстановленные трехмерные координаты объекта. Метод

триангуляции на основе двух камер представляет собой некоторое нелинейное преобразование координат объекта в плоскостях камер в его реальные координаты в сцене. Соответственно, помехи, которые присутствуют в координатах объекта на плоскостях камер, также войдут в данную нелинейную процедуру, что приведет к нелинейности помех в процессе наблюдения. В то же время фильтр Калмана предполагает аддитивность помех как в процессе наблюдения, так и в процессе изменения состояния системы, и будет работать некорректно при невыполнении данных условий. В [16] был предложен несколько другой подход, основанный на использовании ректифицированных изображений полученных со стереопары. Состояние динамической системы было выбрано аналогично упомянутой ранее статье, а в качестве наблюдений рассматривались координаты объекта на одном из изображений и величина смещения между его проекциями на плоскости каждой из камер. Т.к. такой процесс наблюдения является нелинейным, авторы предложили использовать расширенный вариант фильтра Калмана, который решает задачу фильтрации нелинейных динамических систем. Необходимость ректификации в данном подходе несколько ограничивает его применимость. В случаях, когда проекции центров камер попадают в изображения, результаты ректификации этих изображений будут растягиваться на бесконечность. В работах [17,18] в качестве наблюдений использовались координаты проекций объекта на каждую из камер. Такой подход дает достаточно точные оценки, его работоспособность теоретически никак не ограничена взаимным расположением камер, и он правильным образом трактует присутствующий в системе шум.

Фильтрация динамических систем

Как уже упоминалось ранее, движущийся объект можно рассматривать как некоторую динамическую систему с состоянием, включающим в себя положение объекта и какие-либо другие его характеристики (скорость, ускорение). По данной системе производятся зашумленные наблюдения (положение объекта на изображении), на основе которых необходимо производить оценки состояния

объекта в каждый момент времени. В общем случае данный процесс можно рассматривать следующим образом:

$$\begin{aligned}x_k &= f(x_{k-1}, w_k) \\ y_k &= h(x_k, v_k).\end{aligned}$$

Здесь x_k – состояние системы, y_k – наблюдение, w_k, v_k – некоторые помехи, функция $f(x_{k-1}, w_k)$ отражает динамику системы, а $h(x_k, v_k)$ процесс наблюдения. Большинство алгоритмов на каждой итерации, так или иначе, рассматривают плотность условной вероятности $p(x_k | y_1, \dots, y_k)$ [28]. В качестве оценки состояния в текущий момент времени можно использовать различные характеристики данной плотности (математическое ожидание, мода, медиана).

В случае, когда помехи имеют аддитивный характер и являются нормальными случайными процессами с математическим ожиданием, равным нулю, для решения задачи оценивания состояния динамической системы используются различные вариации фильтра Калмана [29]. В стандартном варианте фильтр Калмана рассматривает динамическую систему с линейным процессом изменения состояния и линейной функцией наблюдения:

$$\begin{aligned}x_k &= A_k * x_{k-1} + w_k \\ y_k &= H_k * x_k + v_k.\end{aligned}$$

При этом оценки выбираются исходя из критерия минимизации среднеквадратической ошибки $E[\|x_k - \tilde{x}_k\|^2]$, что эквивалентно минимизации следа матрицы ковариации $P_k = cov(x_k - \tilde{x}_k)$. Сам фильтр представляет собой итеративную процедуру обновления параметров матрицы P_k и состояния x_k . Более подробное описание алгоритма изложено в Приложении I. Кроме стандартного варианта существует расширенный вариант фильтра Калмана, решающий задачу фильтрации систем вида:

$$x_k = f(x_{k-1}) + w_k$$

$$y_k = h(x_k) + v_k.$$

Стохастическая аппроксимация

Задачу фильтрации можно рассматривать как задачу оптимизации некоторого функционала по зашумленным наблюдениям. Для решения такого рода задач в оптимизации используются различные алгоритмы стохастической аппроксимации [30]. В большинстве подходов рассматривается задача минимизации функции вида:

$$f(X) = E_w[F(w, X)],$$

где w – некоторая помеха. Оценки минимума производятся на основе зашумленных измерений значения функционала $F(w, X)$ в выбранных точках. Общий подход к построению оценок можно формализовать в виде следующей процедуры:

$$\hat{\theta}_n = \hat{\theta}_{n-1} - \alpha_n * \hat{g}_n(\hat{\theta}_{n-1}).$$

Здесь $\{\hat{\theta}_n\}$ – последовательность оценок минимума, $\{\alpha_n\}$ – некоторая выбранная последовательность неотрицательных чисел, $\hat{g}_n(\hat{\theta}_{n-1})$ – псевдоградиент функции $f(X)$, который “в среднем” должен быть равен ее градиенту и близок к нулю, когда его аргумент стремится к точке экстремума. В качестве примера процедуры такого вида можно привести алгоритм Кифера-Вольфовица [31]:

$$\hat{\theta}_n = \hat{\theta}_{n-1} - \alpha_n \frac{y_n^+ - y_n^-}{2\beta_n}$$

$$y^- = \begin{pmatrix} y_1^- \\ \dots \\ y_r^- \end{pmatrix} = \begin{pmatrix} F(w_{2r(n-1)+1}, \hat{\theta}_n - \beta_n * e_1) \\ \dots \\ F(w_{2r(n-1)+2r-1}, \hat{\theta}_n - \beta_n * e_r) \end{pmatrix}$$

$$y^+ = \begin{pmatrix} y_1^+ \\ \dots \\ y_r^+ \end{pmatrix} = \begin{pmatrix} F(w_{2r(n-1)+2}, \hat{\theta}_n + \beta_n * e_1) \\ \dots \\ F(w_{2r(n-1)+2r}, \hat{\theta}_n + \beta_n * e_r) \end{pmatrix}.$$

Для доказательства состоятельности оценок данного алгоритма приходится накладывать сильные ограничения на помехи, более того, на каждой итерации приходится производить $2r$ наблюдений, что существенно ограничивает его применимость для многомерных задач. Методы рандомизированной стохастической аппроксимации (SPSA) позволяют производить оценки в условиях почти произвольных аддитивных помех, при этом на каждой итерации необходимо лишь одно или два измерения. Выбирая на каждой итерации некоторое случайное направление Δ_n и используя в качестве аппроксимации градиента приближение его проекции на данное направление, оценка минимума происходит при помощи следующих процедур:

$$y_n^+ = F(w_n^+, \hat{\theta}_{n-1} + \beta_n \Delta_n) + v_n^+$$

$$y_n^- = F(w_n^-, \hat{\theta}_{n-1} - \beta_n \Delta_n) + v_n^-$$

Алгоритм с двумя измерениями:

$$\hat{\theta}_n = \hat{\theta}_{n-1} - \alpha_n \Delta_n \frac{y_n^+ - y_n^-}{2\beta_n}$$

Алгоритм с одним измерением:

$$\hat{\theta}_n = \hat{\theta}_{n-1} - \frac{\alpha_n}{\beta_n} \Delta_n y_n^+$$

В случае задачи фильтрации оптимизируемый функционал и его минимум будут изменяться во времени. Авторы [31] предложили метод, позволяющий отслеживать минимум нестационарного функционала $F(x_n, w_n, n)$ по

зашумленным наблюдениям $y_n = F(x_n, w_n, n) + v_n$. На основе последовательности одновременных пробных возмущений Δ_n оценки минимума $\hat{\theta}_n$ в предложенном алгоритме строятся следующим образом:

$$\begin{aligned}x_{2n} &= \hat{\theta}_{2n-2} + \beta \Delta_n, x_{2n-1} = \hat{\theta}_{2n-2} - \beta \Delta_n \\ \hat{\theta}_{2n} &= \hat{\theta}_{2n-2} - \frac{\alpha}{2\beta} \Delta_n (y_{2n} - y_{2n-1}).\end{aligned}$$

Для задачи фильтрации данный алгоритм может давать неудовлетворительные оценки, т.к. в его процедуре никак не учитывается известный характер изменения состояния динамической системы. Более того, для доказательства состоятельности оценок алгоритма требуется выполнение условия $\|\hat{\theta}_n - \hat{\theta}_{n-1}\| < D$, которое может быть не выполнено для изменения состояния произвольной динамической системы.

Модель наблюдения за объектом на основе двух камер

Математическая модель представления камер

Рассмотрим модель наблюдения за объектом при помощи двух камер. Для математического представления камер чаще всего используется модель камеры-обскура. В данной модели каждая камера представляется некоторой плоскостью и выбранной точкой, называемой центром проектирования. Для получения изображения произвольной трехмерной точки проводится прямая, проходящая через эту точку и центр проектирования. Пересечение этой прямой и плоскости камеры и есть искомое изображение. На Рисунке 1 схематично представлена эта операция. Теперь выберем систему координат таким образом, что оси X и Y выбраны внутри плоскости камеры, а ось Z перпендикулярна ей. Обозначив за f – расстояние между центром проектирования и плоскостью камеры для точки $\{X^1, X^2, X^3\}$ координаты ее изображения $\{x^1, x^2\}$ будут вычисляться следующим образом:

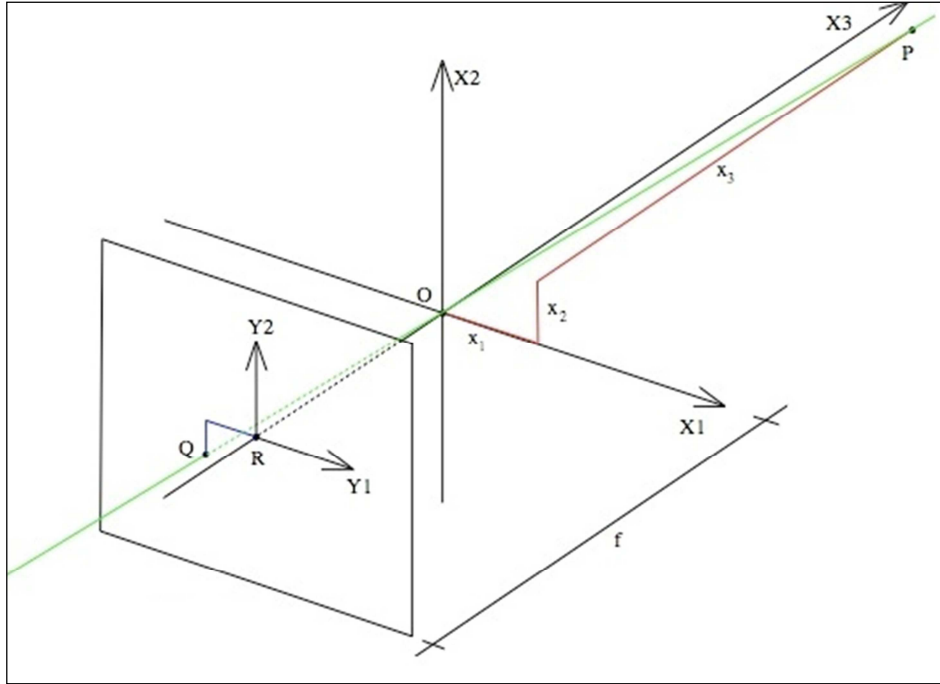


Рис. 1. P – трехмерная точка, O – центр проектирования, Q – изображение точки.

$$x^1 = \frac{fX^1}{X^3}$$

$$x^2 = \frac{fX^2}{X^3}.$$

Преобразования такого рода удобнее всего записывать при помощи использования однородных координат. В пространстве \mathbb{R}^3 можно задать отношение эквивалентности:

$$\{x_1, x_2, x_3\} \sim \{y_1, y_2, y_3\} \Leftrightarrow \exists \omega \in \mathbb{R}, \omega \neq 0 : \{x_1, x_2, x_3\} = \omega \{y_1, y_2, y_3\}$$

Каждому классу эквивалентности можно сопоставить единственным образом вектор из \mathbb{R}^2 . Взяв произвольного представителя класса $\{x_1, x_2, x_3\}$, соответствующий данному классу двухмерный вектор вычисляется как: $\left\{ \frac{x_1}{x_3}, \frac{x_2}{x_3} \right\}$.

Классы, у которых $x_3 = 0$, соответствуют предельным точкам прямых. Такое представление векторов из \mathbb{R}^2 называется однородными координатами. Аналогичным образом можно ввести однородные координаты и для \mathbb{R}^3 . Используя данное представление, вычисление проекции трехмерной точки на

плоскость камеры можно записать как умножение на матрицу:

$$A = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$
$$\bar{x} = A * \begin{pmatrix} X^1 \\ X^2 \\ X^3 \\ 1 \end{pmatrix}$$

В общем случае координаты трехмерной точки не будут заданы в системе координат, связанной с камерой. Поэтому в операцию проектирования необходимо ввести преобразование координат из глобальной системы в систему камеры:

$$K = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$\bar{x} = K[R|t] \begin{pmatrix} X^1 \\ X^2 \\ X^3 \\ 1 \end{pmatrix}$$

Здесь $[R|t]$ представляют сдвиг и поворот, которые делают необходимое преобразование координат. Далее с каждой из камер будет связана матрица $P = K[R|t]$, осуществляющая операцию проектирования.

Определение положения точки на основе двух камер

Рассмотрим пару камер с матрицами P_1 и P_2 . Для произвольной точки \bar{X} , заданной в однородных координатах, и ее проекций \bar{x}_1, \bar{x}_2 на плоскости первой и второй камеры соответственно будут выполнены с точностью до эквивалентности следующие равенства:

$$\bar{x}_1 \cong P_1 \bar{X}$$

$$\bar{x}_2 \cong P_2 \bar{X}$$

Что эквивалентно:

$$\bar{x}_1 \times P_1 \bar{X} = 0$$

$$\bar{x}_2 \times P_2 \bar{X} = 0$$

Последние равенства, при известных координатах проекций $\bar{x}_1 = \{x_1^1, x_1^2, 1\}$ и $\bar{x}_2 = \{x_2^1, x_2^2, 1\}$, можно рассматривать как линейную систему относительно компонент вектора \bar{X} :

$$A\bar{X} = 0$$

$$A = \begin{pmatrix} x_1^1 p_1^{3T} - p_1^{1T} \\ x_1^2 p_1^{3T} - p_1^{2T} \\ x_2^1 p_2^{3T} - p_2^{1T} \\ x_2^2 p_2^{3T} - p_2^{2T} \end{pmatrix}$$

Где p_k^{iT} – это i -ая строка матрицы P_k . Очевидно, что решение данной системы можно найти только с точностью до умножения на некоторый ненулевой скаляр. Поэтому для нахождения вектора \bar{X} необходимо вводить дополнительные ограничения. К примеру, можно искать решение в виде $\bar{X} = \{X, Y, Z, 1\}$, используя метод наименьших квадратов. Но при удалении точки от камер, устойчивость поиска решения таким способом будет падать. Избавиться от данной проблемы, можно рассмотрев систему вида:

$$A\bar{X} = 0$$

$$\|\bar{X}\| = 1$$

Решением такой системы будет правый сингулярный вектор матрицы A , соответствующий минимальному сингулярному числу [33].

Описанный подход к определению положения точки работает крайне нестабильно в условиях зашумленных наблюдений. Уже при невысоком уровне помех в изображении, оценки реального положения точки будут

неудовлетворительными (Рис. 2), что и приводит к необходимости использования дополнительных методов фильтрации.

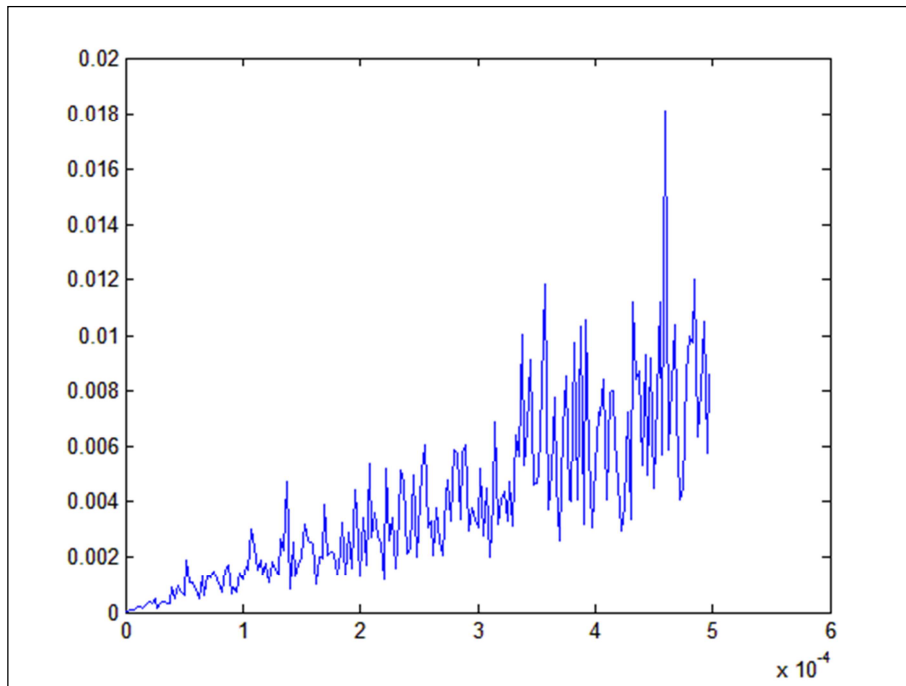


Рис. 2. Рост ошибки определения положения точки в зависимости от уровня погрешности определения ее положения на изображении.

Глава II. Алгоритм фильтрации динамических систем с линейным процессом изменения состояния.

Постановка задачи фильтрации.

Рассмотрим динамическую систему с линейным изменением состояния:

$$x_k = A * x_{k-1} + w_k$$

По ней производятся измерения, характеризующиеся нелинейной функцией h :

$$y_k = h(x_k) + v_k$$

Здесь $x_k \in \mathbb{R}^n$, $y_k \in \mathbb{R}^m$. В каждый момент времени необходимо производить оценки состояния x_k таким образом, что получаемая траектория будет согласована с наблюдениями наилучшим образом. Данную задачу оценивания можно сформулировать как задачу нестационарной оптимизации для некоторого функционала $F(x_k, v_k, k)$. Исходя из логики минимизации ошибки наблюдения, выберем функционал следующим образом:

$$F(x_k, v_k, k) = \|h(x_k) - y_k\|^2.$$

Для отслеживания минимума данного функционала можно использовать метод нестационарной стохастической оптимизации, описанный в [31]. Но т.к. выбранная функция качества никак не отражает динамику изменения состояния системы, то оценки, производимые алгоритмом, будут неудовлетворительными. Поэтому в процедуру оптимизации необходимо ввести модификации, учитывающие характер изменения состояния системы.

Модифицированный алгоритм нестационарной стохастической оптимизации

Рассмотрим последовательность векторов $\{\Delta_k\}$, представляющих собой одновременные пробные возмущения. Каждый вектор является реализацией некоторого случайного вектора размерности n , компоненты которого могут равновероятно принимать значения $\pm \frac{1}{\sqrt{n}}$. Имея некоторое начальное приближения состояния системы $\hat{\theta}_0$, последовательность оценок будем строить

следующим образом:

Алгоритм 1

$$\begin{aligned}
 x_{2k} &= A^2 * [\hat{\theta}_{2k-2} + \beta \Delta_k] \\
 x_{2k-1} &= A * [\hat{\theta}_{2k-2} - \beta \Delta_k] \\
 z_k &= \|h(x_k) - y_k\|^2 \\
 \hat{\theta}_{2k} &= A^2 * \hat{\theta}_{2k-2} - \frac{\alpha}{2\beta} \Delta_k (z_{2k} - z_{2k-1}) \\
 \hat{\theta}_{2k-1} &= A * \hat{\theta}_{2k-2}.
 \end{aligned}$$

Здесь α и β – положительные числа, являющиеся параметрами алгоритма. Данная процедура основана на алгоритме, представленном в [31]. На одной итерации алгоритма производится оценки сразу для двух состояний, такой подход позволяет более точно приближать градиент функции, т.к. используется информация о наблюдениях в различные моменты времени. Точки x_{2k} и x_{2k-1} выбираются таким образом, что в оценке градиента функции будет учтена динамика изменения состояния системы. Также, для получения новой оценки, аналогично фильтру Калмана, производится операция предсказания эволюции старой оценки при помощи матрицы A . Такая структура алгоритма позволяет ему адаптироваться к изменению состояния системы и за счет этого делать более точные оценки. Теперь рассмотрим, каким образом можно применить данный алгоритм к задаче отслеживания объекта при помощи двух камер.

Отслеживание объекта при помощи двух камер

Рассмотрим пару камер, которые наблюдают движущуюся точку. Предположим, что известны внутренние и внешние параметры камер в виде матриц проектирования P_1 , P_2 . Также предположим, что точка движется с некоторой постоянной скоростью. Тогда в качестве состояния динамической системы выберем вектор, состоящий из положения точки и ее скорости:

$$x_k = [x_k^1 \quad x_k^2 \quad x_k^3 \quad v_{x_k}^1 \quad v_{x_k}^2 \quad v_{x_k}^3]^T$$

Динамика системы будет выглядеть следующим образом:

$$x_k = [x_{k-1}^1 + v_{x_{k-1}}^1 \quad x_{k-1}^2 + v_{x_{k-1}}^2 \quad x_{k-1}^3 + v_{x_{k-1}}^3 \quad v_{x_{k-1}}^1 \quad v_{x_{k-1}}^2 \quad v_{x_{k-1}}^3]^T,$$

что в матричном виде выглядит как:

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x_k = A * x_{k-1}$$

В качестве наблюдения выберем изображения точки на каждой из камер. Для этого рассмотрим функцию $projection(x, P)$, которая для трехмерной точки x и матрицы P некоторой камеры, возвращает проекцию точки на плоскость этой камеры. Тогда в качестве функции $h(x_k)$ из модели динамической системы возьмем следующее:

$$h(x_k) = \begin{bmatrix} projection(x_k, P_1) \\ projection(x_k, P_2) \end{bmatrix}$$

Таким образом, задача отслеживания объекта сформулирована в терминах динамической системы. Теперь для оценивания положения объекта можно использовать алгоритм, описанный в предыдущем разделе. Функция качества $\|h(x_k) - y_k\|^2$ в данном контексте представляет собой суммарное отхождение проекций трехмерной точки на плоскости каждой из камер от произведенного наблюдения. Чаще всего ее называют ошибкой перепроектирования. Необходимо заметить, что данная функция является локально выпуклой и гладкой. Согласно [32] в условии помех она может иметь большое число точек экстремума, что существенно осложняет решение задачи поиска ее минимума методами локальной оптимизации. Для корректной работы алгоритма необходимо правильным образом выбирать его параметры и ограничивать область поиска минимума, используя операцию проектирования градиента функции.

Глава III. Практические результаты.

Тестирование алгоритма.

Для начала, рассмотрим работу алгоритма нестационарной стохастической аппроксимации из статьи [31] применительно к задаче отслеживания объекта. Как уже говорилось ранее, данный алгоритм не учитывает динамику движения точки, что может привести к сходимости его оценок к неправильной траектории. На Рис. 3 показан пример работы данного алгоритма в условии отсутствия помех. На левом графике зеленым цветом отображена реальная траектория движения точки, синим – оценки алгоритма. Видно, что алгоритм правильно находит направление, в котором движется точка, но его оценки «защелкиваются» примерно на середине траектории.

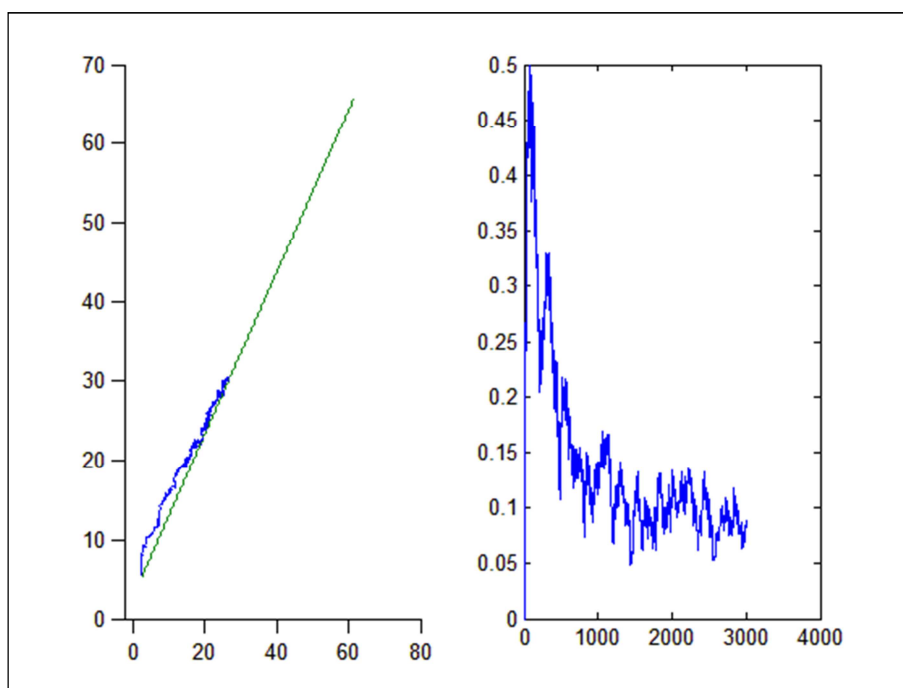


Рис. 3. Левый график: реальная траектория и оценки алгоритма. Правый график: ошибка наблюдения в зависимости от номера итерации.

Стоит заметить, что ошибка наблюдения (правый график) падает с течением времени и стабилизируется, начиная с определенного момента. В целом, такое поведение алгоритма объясняется спецификой оптимизируемой

функции, а именно, ее исключительно локальной выпуклостью. Оценки алгоритма попадают в область одного из локальных минимумов, что приводит к получению неправильной траектории движения.

Теперь рассмотрим работу модифицированного алгоритма в аналогичных условиях, т.е. при отсутствии помех. На Рис. 4 видно, что оценки модифицированного алгоритма идеально совпадают с реальной траекторией движения, при этом присутствие ошибки наблюдения связано исключительно со случайным характером процедуры оценивания. За счет использования информации о динамике системы в условии отсутствия помех алгоритму нет необходимости производить какую-либо существенную коррекцию оценок.

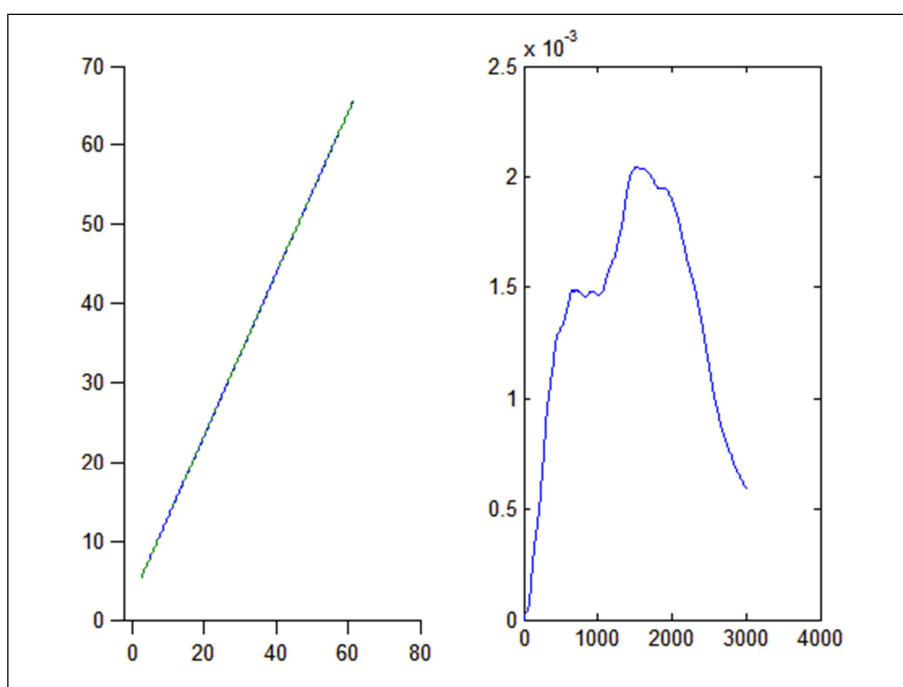


Рис. 4

При добавлении помех во время работы алгоритма может возникнуть ситуация сильного отхождения оценок от правильной траектории. При этом отойдя от искомой траектории на слишком большое расстояние, алгоритм уже не сможет вернуться к ней. Такие ситуации возникают вследствие того, что при удалении точки от камеры, изображение траектории ее движения с течением времени будет сходиться в некоторую точку (Рис. 5). Из-за этого у функции качества будут присутствовать области с почти постоянным значением. Попав в такую область, траектория, получаемая алгоритмом, выйти из нее уже не сможет

из-за свойств локальной оптимизации.

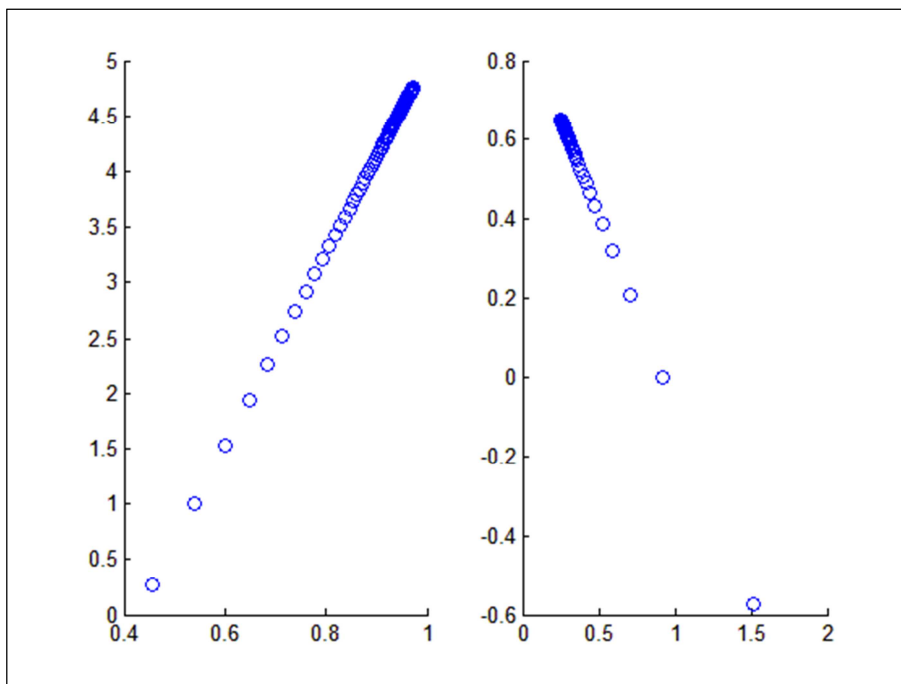


Рис. 5. Изображения траектории движения точки на плоскостях двух камер.

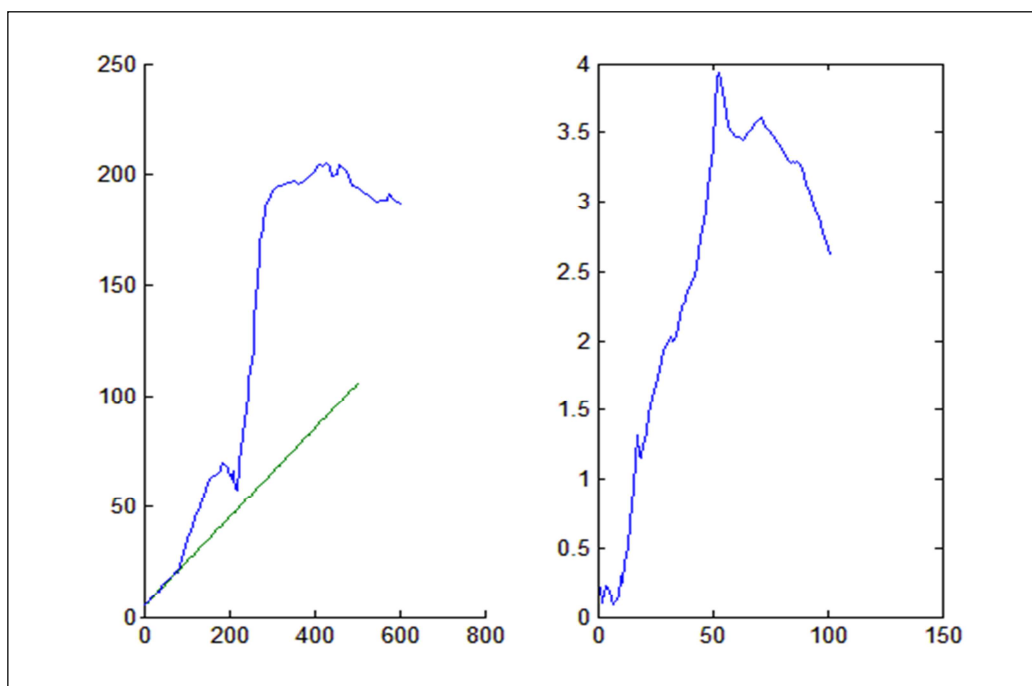


Рис. 6. Результаты работы модифицированного алгоритма без применения операции проектирования градиента.

На Рис. 4 показан пример такого случая. В нем оценки алгоритма с определенного момента времени начинают сильно отходить от реальной траектории, что в итоге приводит к их постоянному ухудшению. Для решения

этой проблемы при реализации алгоритма необходимо ограничивать величину, на которую может сдвигаться оценка. На каждой итерации алгоритм вычисляет величину $d = \frac{z_{2k} - z_{2k-1}}{2\beta}$, которая задает размер сдвига оценки в случайном направлении. Добавив в процедуру простое правило проектирования:

$$if(abs(d) > th) d = sign(d) * th$$

можно избавиться от проблемы выхода траектории за пределы области, в которой находится искомый минимум.

Рассмотрим результаты работы алгоритма при различных уровнях помех. Обозначим за σ^{img} уровень ошибки в наблюдениях (Рис. 7), а за σ^{start} уровень ошибки в начальном приближении.

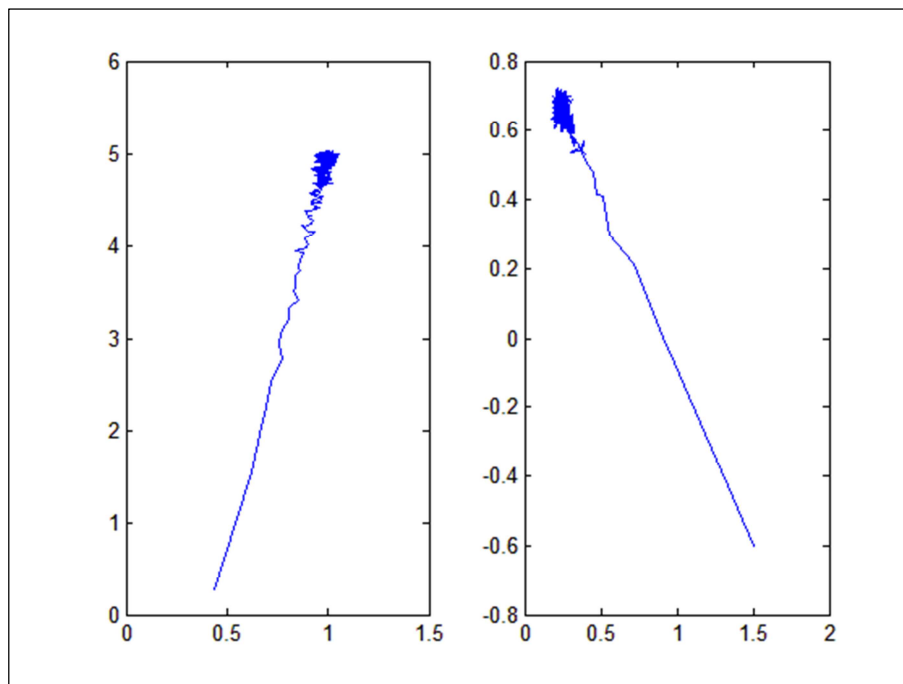


Рис. 7. Пример зашумленных наблюдений при $\sigma^{img} = 0,02$. Графики представляют собой изображения на плоскостях камер.

На Рис. 8 и 9 представлена работа алгоритма с различными параметрами шума. На данных примерах видно, что оценки с течением времени становятся менее точными, что объясняется удалением точки от камер. При увеличении

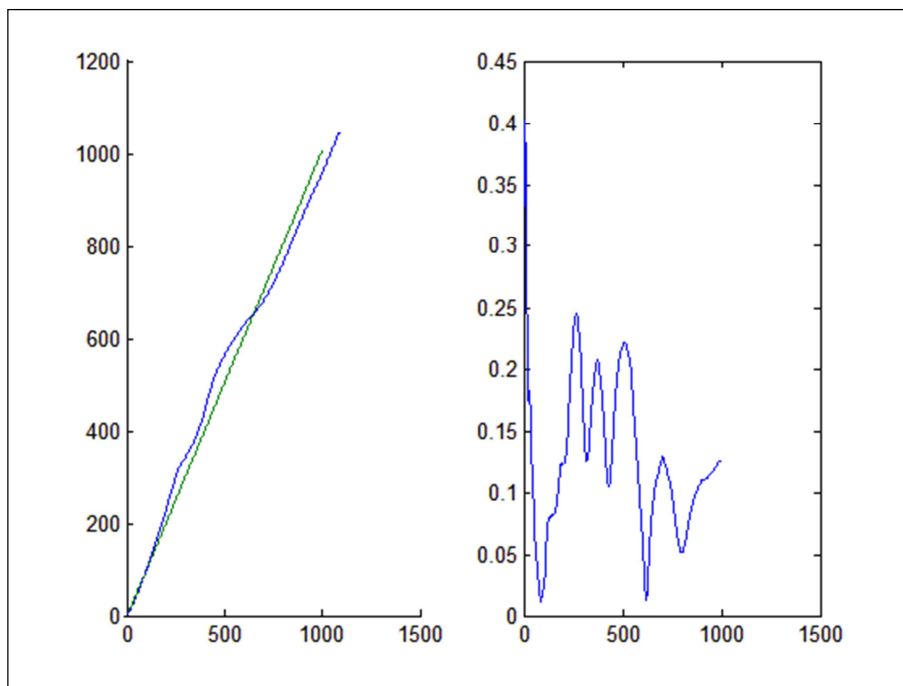


Рис. 8. $\sigma^{img} = 0,02$ $\sigma^{start} = 0,5$

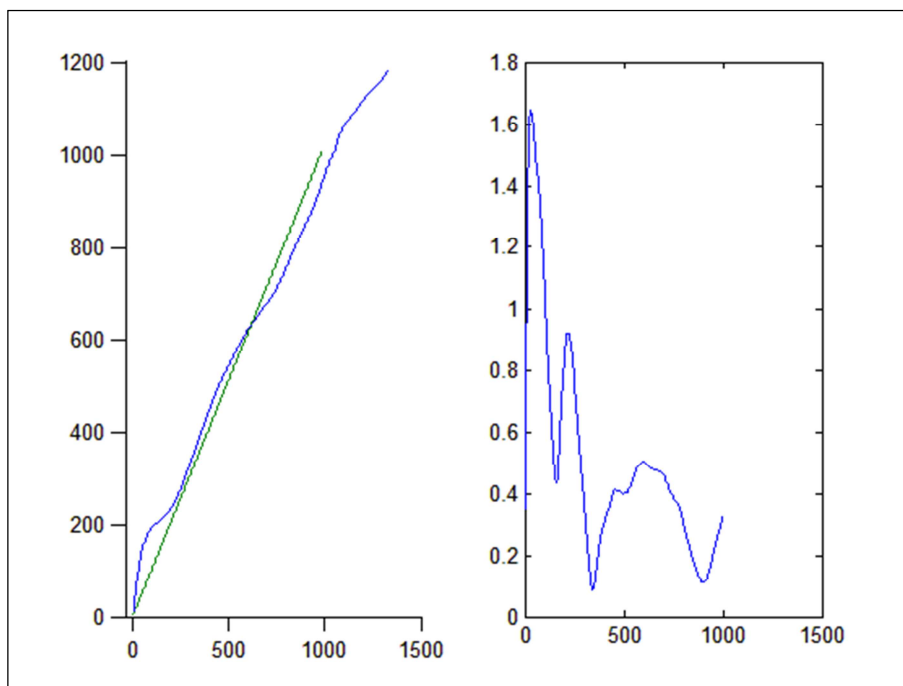


Рис. 9. $\sigma^{img} = 0,1$ $\sigma^{start} = 1,0$

расстояния от точки до камеры, возможная точность восстановления ее положения методами минимизации ошибки перепроектирования будет падать. Ошибка наблюдения в каждом из примеров на первых итерациях убывает, после чего стабилизируется в некоторой области. Снижение ошибки наблюдения происходит за счет подавления ошибки в начальном приближении.

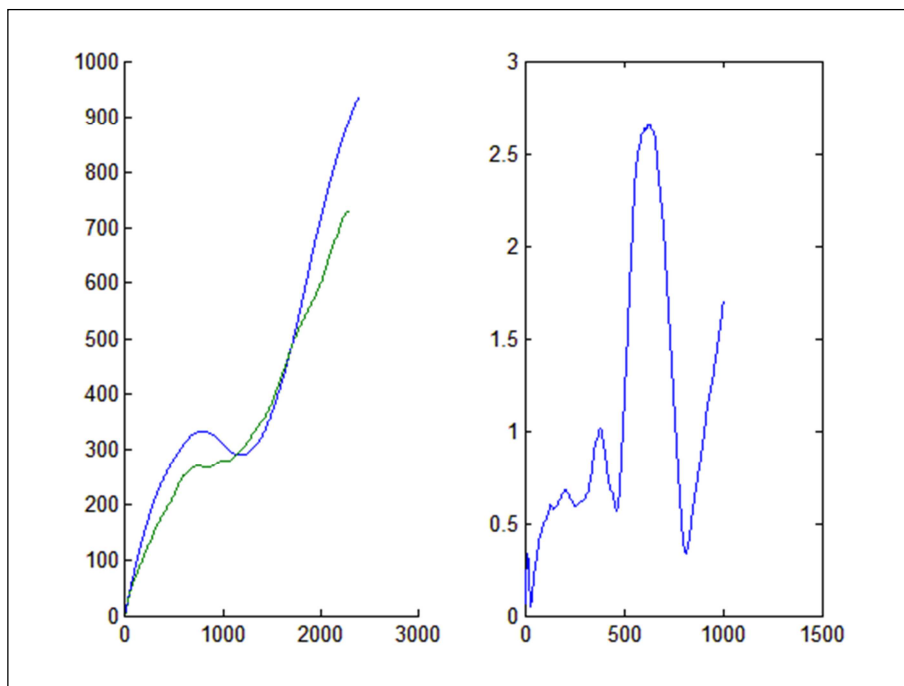


Рис. 9. Пример работы алгоритма с шумом в динамике системы.

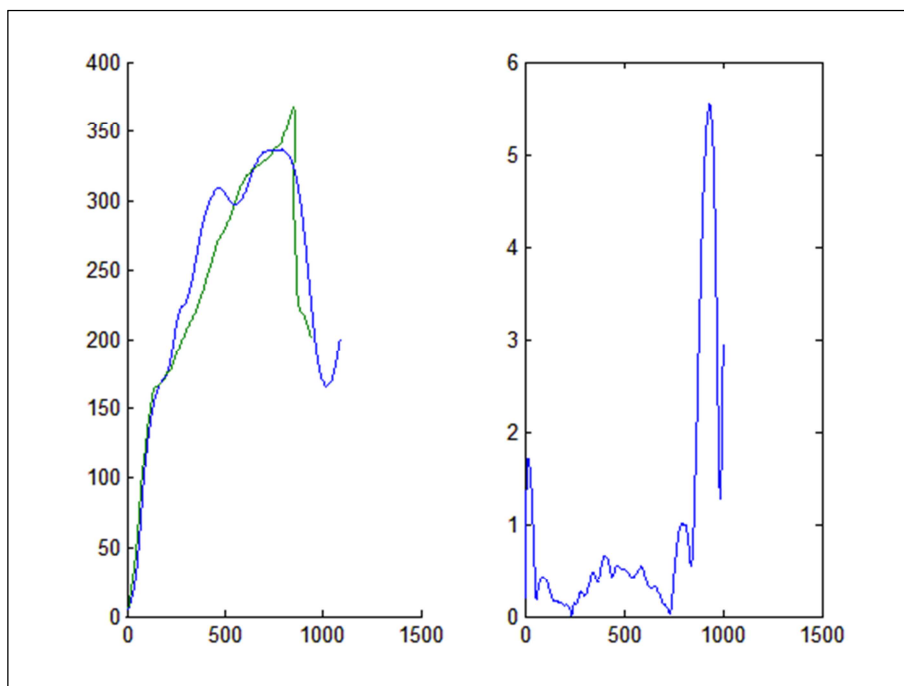


Рис. 10

Теперь добавим помеху в динамику системы. Реальные траектории при этом уже не будут прямолинейными, а будут иметь произвольный вид. На Рис. 9 и 10 видно, что алгоритм в такой ситуации успешно адаптируется к изменению направления движения, но работает менее стабильно, чем в случае отсутствия помехи в динамике системы.

Сравнение с фильтром Калмана

Рассмотрим работу нелинейного фильтра Калмана в аналогичной модели наблюдения за движущейся точкой. На Рис. 11 и 13 представлены примеры работы фильтра при различных условиях шума.

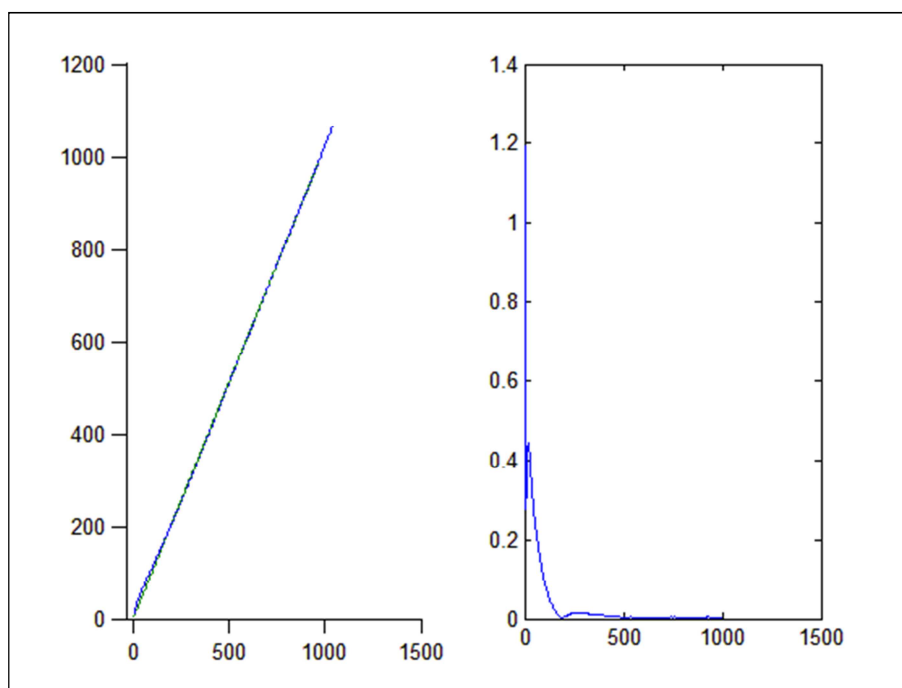


Рис. 11. $\sigma^{img} = 0,02$ $\sigma^{start} = 0,5$

Фильтр Калмана дает более точные оценки для траектории движения объекта. Также ошибка наблюдения более стабильна и значительно меньше, чем в результатах Алгоритма 1. Несмотря на это, использование фильтра Калмана не всегда возможно. Для корректной работы фильтра необходимо правильным образом выбрать матрицы ковариации помех, что часто бывает затруднительно при отсутствии какой-либо информации о характере и величине шума. Если значения матриц ковариации не соответствует реальным характеристикам шума, то фильтр Калмана может давать совершенно неправильные оценки. На Рис. 12 показан пример такого случая, матрицы ковариаций были заданы неверно, что привело к неадекватным оценкам. В то же время параметры Алгоритма 1 никак не зависят от характеристик шума. Выбор параметра β зависит только от характеристик выпуклости оптимизируемой функции и желаемой точности

получаемых оценок, параметр α представляет собой стандартный размер шага для алгоритмов градиентного спуска, и для его выбора можно применять различные адаптивные процедуры разработанные в области локальной оптимизации [34].

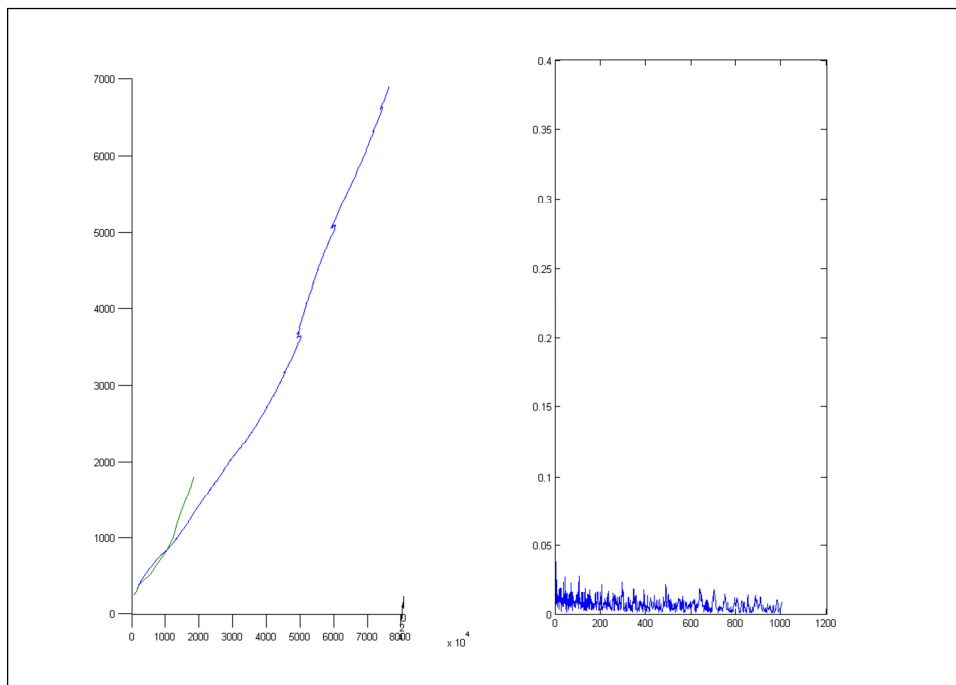


Рис. 12. Результат работы фильтра Калмана при неправильных параметрах матриц ковариаций.

Также стоит заметить что, процедура фильтра Калмана содержит в себе такие сложные операции, как обращение матриц и вычисление градиента векторной функции, вследствие чего фильтр работает значительно медленней Алгоритма 1. Ниже представлена таблица, в которой указано время в секундах, затраченное, на фильтрацию траектории каждым из алгоритмов. Преимущество в скорости работы будет еще более существенным при повышении размерности задачи или фильтрации траекторий множества точек.

Количество итераций	Фильтр Калмана	Алгоритм 1
1000	0,621511	0,030874
3000	1,909988	0,128089
5000	3,213213	0,267196

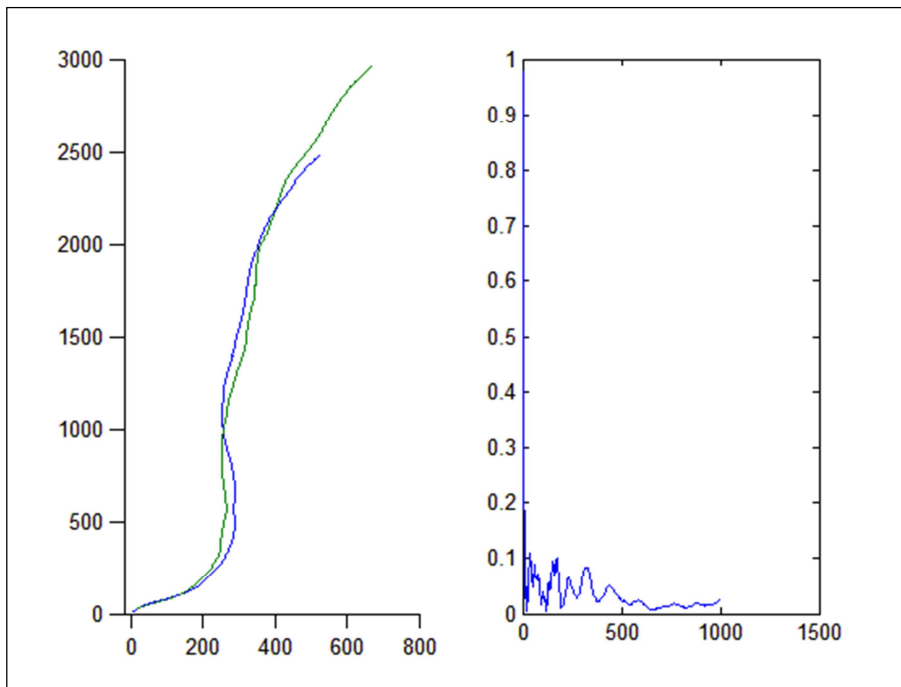


Рис. 13. Результат работы фильтра Калмана для случая помехи в динамике системы.

Заключение

В данной работе был представлен алгоритм отслеживания объекта по его изображению на двух камерах на основе метода нестационарной стохастической оптимизации. Разработанный алгоритм применим не только к задаче отслеживания объекта, но и к произвольной задаче фильтрации динамической системы с линейным процессом изменения состояния. Для тестирования алгоритма была реализована модель, симулирующая работу двух камер, которые наблюдают движение точки. Были проведены тесты алгоритма с различными условиями помех и вариантами траекторий движения точки. Проведен анализ результатов тестирования, на основании которого можно сделать вывод о том, что алгоритм успешно справляется с задачей отслеживания траектории движения точки и минимизации ошибки наблюдения для получаемых оценок. В ходе тестирования были получены результаты о том, что не модифицированный алгоритм нестационарной стохастической аппроксимации [31] не справляется с задачей отслеживания движущегося объекта даже в условиях отсутствия помех. Сравнительный анализ работы нового алгоритма и фильтра Калмана показал, что фильтр Калмана дает более точные оценки, но в то же время требует настройки большего числа параметров, которые к тому же зависят от характеристик шума. Кроме того были получены результаты о том, что новый алгоритм работает значительно быстрее фильтра Калмана, что позволяет использовать его для фильтрации систем большей размерности или же для одновременной фильтрации нескольких систем. Моделирование показало работоспособность нового алгоритма в задачах, где точность оценок не является критической, а вычислительных ресурсов ограниченное количество, например - избежание препятствий автономно перемещающимся роботом

Список литературы

1. *R. Cucchiara, C. Grana, M. Piccardi, and A. Prati.* Detecting moving objects, ghosts and shadows in video streams. *IEEE Trans. on Patt. Anal. and Machine Intell.*, vol. 25, no. 10, Oct. 2003, pp. 1337-1342
2. *B.P.L. Lo and S.A. Velastin.* Automatic congestion detection system for underground platforms. *Proc. of 2001 Int. Symp. on Intell. Multimedia, Video and Speech Processing*, pp. 158-161, 2000
3. *C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland.* *Pfinder:Real-time Tracking of the Human Body.* *IEEE Trans. on Patt. Anal. and Machine Intell.*, vol. 19, no. 7, pp. 780-785, 1997.
4. *C. Stauffer, W.E.L. Grimson.* Adaptive background mixture models for real-time tracking. *Proc. of CVPR 1999*, pp. 246-252.
5. *Elgammal, A., Harwood, D., and Davis, L.S.* Non-parametric Model for Background Subtraction. *Proc. of ICCV '99 FRAME-RATE Workshop*, 1999.
6. *Lowe, David G.* Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision. 2.* pp. 1150–1157
7. *Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool.* SURF: Speeded Up Robust Features, *Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, pp. 346--359, 2008
8. *Duy-Nguyen Ta, Wei-Chao Chen, Natasha Gelfand, Kari Pulli.* SURFTrac: Efficient tracking and continuous object recognition using local feature descriptors. *CVPR 2009*: 2937-2944
9. *Viola and Jones.* Rapid object detection using boosted cascade of simple features. *Computer Vision and Pattern Recognition*, 2001
10. *T.Mita, T. Kaneko, and O. Hori.* Joint Haar-like features for face detection. In *ICCV*, 2005.
11. *Sungji Han, Youngjoon Han and Hernsoo Hahn.* Vehicle Detection Method using Haar-like Feature on Real Time System. *World Academy of Science, Engineering and Technology* 59 2009.

12. *Yoav Freund, Robert E. Schapire.* A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting, 1995
13. *S. Rougeaux, N. Kita, Y. Kuniyoshi, S. Sakane.* Tracking A Moving Object With A Stereo Camera Head. In Proc. 11th Annual Conf. of Robotics Society of Japan.
14. *L. Ho Gi Jung, Yun Hee Lee, Dong Suk Kim, Pal Joo Yoon.* Stereo Vision Based Advanced Driver Assistance System.
15. *Ivana Mikic, Simone Santini, Ramesh Jain.* Video Processing and Integration from Multiple Cameras. In Proceedings of the 1998 Image Understanding Workshop, Morgan-Kaufman.
16. *Hernan Badino, Uwe Franke, Clemens Rabe, Stefan Gehrig, Daimlerchrysler Ag.* Stereo Vision-Based Detection of Moving Objects Under Strong Camera Motion.
17. *Shaojie She.* Navigation System for Autonomous Mobile Robot - A Simultaneous localization and Mapping Approach.
18. *Gabe Sibley, Gaurav Sukhatme, Larry Matthies.* The Iterated Sigma Point Kalman Filter with Applications to Long Range Stereo. Robotics: Science and Systems II.
19. *Shoushtarian, B., Bez, H.E.* Object Tracking Using Background Subtraction and Motion Estimation in MPEG Videos, PRL(26), No. 1, 1 January 2005, pp. 5-26.
20. *B. Boufama and M. A. Ali.* Tracking Multiple People for Video Surveillance, Image Analysis and Recognition Lecture Notes in Computer Science, 2007, Volume 4633/2007, 581-592
21. *Bruce D. Lucas and Takeo Kanade.* An Iterative Image Registration Technique with an Application to Stereo Vision. Proceedings of the 1981 DARPA Image Understanding Workshop, April, 1981, pp. 121-130.
22. *Rosenblatt M.* Remarks on some nonparametric estimates of a density function. Annals of Mathematical Statistics 27: 832–837.
23. *Parzen E.* On estimation of a probability density function and mode. Annals of Mathematical Statistics 33: 1065–1076.

24. *Bhattacharyya A.* On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society* 35: 99–109.
25. *C. Yang, R. Duraiswami, A. Elgammal.* Real-Time Kernel-Based Tracking in Joint Feature-Spatial Spaces. Tech. Rep. CS-TR-4567, University of Maryland, 2004.
26. *Baoxin Li, Chellappa, R., Qinfen Zheng, Der, S.Z.* Model-Based Temporal Object Verification Using Video. *Image Processing, IEEE Transactions*, Issue Date: Jun 2001, Volume: 10 Issue:6
27. *A. Yilmaz, X. Li, M. Shah.* Object contour tracking using level sets. *Asian Conference on Computer Vision, ACCV 2004, Jaju Islands, Korea, 2004*
28. *M. Ribeiro.* Kalman and extended Kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics*, p. 43, 2004
29. *R. E. Kalman.* A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME, Journal of Basic Engineering*, Pg. 35-45, March 1960
30. *Граничин О.Н., Поляк Б.Т.* Рандомизированные алгоритмы оценивания и оптимизации при почти произвольных помехах. М.: Наука. 2003. 291с.
31. *Вахитов А.Т., Граничин О.Н., Гуревич Л.С.* Алгоритм стохастической аппроксимации с пробным возмущением на входе в нестационарной задаче оптимизации. *Автоматика и телемеханика*, 2009, № 11. С. 70-79.
32. *Fredrik Kahl, Richard Hartley.* Multiple-View Geometry under the L1-Norm. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, VOL. 30, NO. 9, SEPTEMBER 2008
33. *Andrew Zisserman, Richard Hartley.* Multiple View Geometry in Computer Vision
34. *M.N. Vrahatisa, G.S. Androulakisa, J.N. Lambrinosa, G.D. Magoulasb.* A class of gradient unconstrained minimization algorithms with adaptive stepsize. *Journal of Computational and Applied Mathematics* 114 (2000) 367-386

Приложение I. Фильтр Калмана.

Фильтр Калмана решает задачу оценивания состояния линейной динамической системы:

$$x_k = A_k * x_{k-1} + w_k$$

$$y_k = H_k * x_k + v_k$$

Оценка на каждом шаге производится исходя из критерия минимизации среднеквадратической ошибки $E[\|x_k - \tilde{x}_k\|^2]$, что эквивалентно минимизации следа матрицы ковариации ошибки $P_k = cov(x_k - \tilde{x}_k)$. Предполагая, что w_k и v_k – нормальные случайные процессы с нулевым математическим ожиданием и матрицами ковариации Q_k и R_k соответственно, оценка на текущем шаге вычисляется при помощи двухэтапной операции:

Этап экстраполяции:

$$\hat{x}_k = A_k * \tilde{x}_{k-1}$$

$$\hat{P}_k = F_k * \tilde{P}_{k-1} * F_k^T$$

Этап коррекции:

$$\hat{z}_k = y_k - H_k * \hat{x}_k$$

$$S_k = H_k * \hat{P}_k * H_k^T + R_k$$

$$K_k = \hat{P}_k * H_k^T * S_k^{-1}$$

$$\tilde{x}_k = \hat{x}_k + K_k * \hat{z}_k$$

$$\tilde{P}_k = (I - K_k * H_k) * \hat{P}_k$$

Для нелинейного случая существует расширенный вариант фильтра Калмана. Для системы:

$$x_k = f(x_{k-1}) + w_k$$

$$y_k = h(x_k) + v_k$$

При тех же предположениях о помехах оценки производятся при помощи

следующей процедуры:

Этап экстраполяции:

$$\hat{x}_k = f(\tilde{x}_k)$$

$$\hat{P}_k = F_{k-1} * \tilde{P}_{k-1} * F_{k-1}^T$$

Этап коррекции:

$$\hat{z}_k = y_k - h(\hat{x}_k)$$

$$S_k = H_k * \hat{P}_k * H_k^T + R_k$$

$$K_k = \hat{P}_k * H_k^T * S_k^{-1}$$

$$\tilde{x}_k = \hat{x}_k + K_k * \hat{z}_k$$

$$\tilde{P}_k = (I - K_k * H_k) * \hat{P}_k$$

$$F_{k-1} = \left. \frac{\partial f}{\partial x} \right|_{\tilde{x}_{k-1}}$$

$$H_k = \left. \frac{\partial h}{\partial x} \right|_{\hat{x}_k}$$