

Реализация данной системы предполагается с использованием механизма нейронных сетей. Однако для этого еще предстоит решить несколько задач, а именно, выбор и описание существенных параметров предметной области, выявление взаимосвязи между параметрами, оптимизация, выбор структуры и характеристик нейронной сети для реализации системы.

Несмотря на это, следует отметить, что в дальнейшем могут быть созданы интеллектуальные системы с нервно-системной организацией структуры, функций и поведения, в основе которых будут лежать рассмотренные принципы.

Список литературы: 1. *Станкевич Л.А.* Когнитивные нейробиологические системы управления. Проблемы нейрокибернетики. Материалы XII Международной конференции по нейрокибернетике. Ростов-на-Дону, 1999. 2. *Ларичев О.И., Петровский А.В.* Системы поддержки принятия решений. Современное состояние и перспективы их развития //Итоги науки и техники. Сер.Техническая кибернетика. Т.21. М. ВИНТИ, 1987. 3. *Мушик, Мюллер.* Методы принятия технических решений. М.: Мир, 1992.

Поступила в редколлегию 28.08.2007

Головий (Гусарь) Наталья Владимировна, аспирантка кафедры Информатики ХНУРЭ. Научные интересы: системный анализ данных. Адрес: Украина, 61166, Харьков, пр.Ленина,14, тел. (057) 702 15 15, e-mail: rica1982@mail.ru.

Ясер Даюб, аспирант кафедры Информатики ХНУРЭ. Научные интересы: системный анализ данных. Адрес: Украина, 61166, Харьков, пр.Ленина,14, тел. (057) 702 15 15, e-mail: kuzy@kture.kharkov.ua.

УДК 004.896, 004.932

И.В. ГАРЯЧЕВСКАЯ

ТЕХНОЛОГИЯ РАЗРАБОТКИ ПО СТЗ, ТЕСТИРОВАНИЯ И АДАПТАЦИИ К УСЛОВИЯМ ЭКСПЛУАТАЦИИ

При создании систем технического зрения (СТЗ) одной из основных задач является выбор методов обработки изображений и их порядка. При этом большое значение имеет стыковка аппаратного (вычислительная техника) и программного (алгоритм обработки изображений) обеспечения. Технология разработки, тестирования и адаптации ПО СТЗ к условиям эксплуатации позволит решить проблему стыковки без этапа натурного испытания.

Введение

СТЗ с точки зрения ПО – это алгоритм обработки изображения некоторой последовательностью методов. На рис.1 приведены возможные этапы обработки изображений для задачи мониторинга пассажиропотока.

Для каждой конкретной задачи набор этих этапов индивидуален, определение этапов является сложной задачей. Кроме этого, существует еще одна проблема, а именно выбор одного метода, наиболее подходящего для каждого из этапов.

Сроки, за которые производится разработка СТЗ, исчисляются месяцами, требуют больших материальных затрат, и обязательных натурных испытаний. Результаты натурных испытаний могут выявить недостатки разработанного ПО и



Рис. 1. Этапы обработки изображений

потребуется улучшений, а это дополнительные временные и материальные затраты.

Кроме этого, даже незначительные изменения условий эксплуатации могут привести к ошибкам работы СТЗ. Сократить время, требуемое на создание ПО СТЗ, и стоимость разработки может предложенная технология.

Анализ основных достижений и публикаций

В настоящее время машинное моделирование – единственный способ анализа и синтеза сложных систем, к которым можно, в частности, отнести и СТЗ. Моделирование используется при проектировании, создании, внедрении, эксплуатации систем, а также на различных уровнях их изучения – начиная от анализа работы элементов и кончая исследованием систем в целом в их взаимодействии с окружающей средой. Ранее уже проводились исследования в области создания технологии разработки ПО СТЗ [1, 2].

Целью проводимого исследования является создание технологии разработки ПО СТЗ с этапом адаптации к условиям эксплуатации.

Методы решения и полученные результаты

Первый шаг технологии – определение структуры СТЗ, т.е. определение необходимых этапов и допустимых методов, реализуется с использованием продукционно-фреймовой модели представления данных.

Комбинация двух известных подходов в один обусловлена достоинствами и недостатками каждой из них.

Так, языком представления знаний целесообразно использовать фреймовую систему, так как фреймовая иерархия базы знаний позволяет знания о методах представить в структурированном виде с сохранением свойства наследования, что является актуальным. Недостатками фреймовой системы является невозможность организовать гибкий механизм логического вывода. А для представления модели предметной области необходима гибкая и простая система вывода. Модель предметной области будет взаимодействовать с методами посредством выводов продукционных правил.

Продукционно-фреймовая модель позволяет сократить объем памяти, необходимый для хранения данных.

Разработанная продукционно-фреймовая модель, проанализировав задачи и функции СТЗ, определит этапы обработки, их последовательность и множество методов для каждого из этапов.

На рис. 2. представлены два алгоритма. Оба включают в себя по одному методу для каждого этапа. Первый алгоритм состоит из метода 1 для первого этапа, метода 1 для второго этапа, метода 1 для третьего этапа. Второй алгоритм состоит из метода 1 для первого этапа, метода 2 для второго этапа, метода 1 для третьего этапа. Таким же образом строятся все остальные алгоритмы.

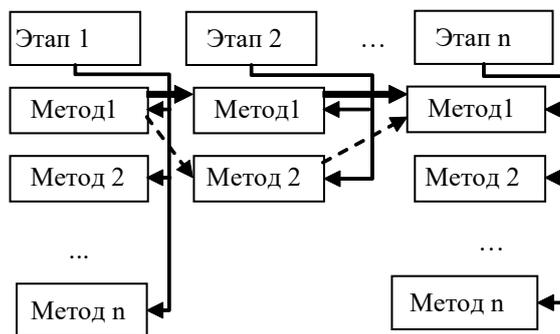


Рис.2. Общий вид формирования алгоритма

Результатом первого шага технологии будет множество алгоритмов, полученных полным перебором из выбранных методов, которые принадлежат выделенному множеству.

Второй шаг технологии – сокращение количества полученных алгоритмов.

Необходимо осуществить выбор из множества сгенерированных алгоритмов одного наиболее удачного.

В общем случае следует получить конкретное решение через модель многокритериального оценивания частных критериев, характеризующих альтернативы, и субъективной информации о предпочтении частных критериев:

$$Q \rightarrow P(a) \rightarrow a^0, \quad P(a) = F[w_i, k_i(a)].$$

Выбор осуществляется по следующему алгоритму:

1. Определить важность (вес) каждого критерия w . (В зависимости от задач, стоящих перед СТЗ, разработчик определяет, какие критерии более важны).

Так, для СТЗ, работающих в режиме реального времени, скорость обработки является ключевым критерием, а для СТЗ в медицине качество будет гораздо более важным критерием по сравнению со временем.

К основным критериям качества работы алгоритма обработки видеoinформации и, как следствие, всей системы технического зрения можно отнести (Таблица):

- величину отклонения между фактическим и контрольным временем обработки кадра;
- количество шума до и после фильтрации;
- расположение границ должно быть как можно ближе к правильным;

| | k1 | k2 | ... | kn | Результирующий показатель |
|-----|-----|-----|-----|-----|---------------------------|
| | w1 | w2 | ... | wn | |
| A1 | z11 | z12 | ... | z1n | A1' |
| A2 | z21 | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... |
| An | zm1 | ... | ... | zmn | ... |

- количество пикселей, выражающих границы объекта, должно быть близким к единице;
- только один результат для одной границы;
- количество ошибок при распознавании и др.

2. Обработать тестовое изображение всеми алгоритмами.

3. Заполнить таблицу значениями по критериям.

4. Удалить те алгоритмы, в которых есть превышение значения хотя бы по одному критерию.

По каждому из критериев существует свое допустимое значение, превышение которого делает невозможным применение алгоритма, в котором использован метод выдавший значение выше допустимого. Например, если при подавлении шума было получено значение 50%, можно точно утверждать, что использовать такой метод нецелесообразно, и как следствие, будет удален весь алгоритм.

По окончании данного шага алгоритма можно предположить, что любой из оставшихся алгоритмов может быть использован в СТЗ, однако необходимо определить наиболее хороший.

5. В оставшихся алгоритмах нормализовать значения:

$$k_i \in [k_{i_{\max}}, k_{i_{\min}}] \quad i = 1, 2, \dots, n \quad ,$$

$$k_{i_{\max}} = 1, \quad k_{i_{\min}} = 0, \quad 0 \leq k_i \leq 1 \quad .$$

Каждый из критериев имеет свою систему исчисления, по критерию 1 будут получены значения в миллисекундах, а по критерию 2 – в процентах. Следовательно, необходимо их привести к общему виду. Выбрав по каждому критерию наиболее хорошее значение, приравняем его к 1, а наиболее плохое значение приравняем к 0. Все промежуточные значения будут расположены соответственно в интервале от 1 до 0.

6. Вычислить результирующий показатель алгоритма.

Вычисление для 1 алгоритма:

$$A_1' = z_{11} \cdot w_1 + z_{12} \cdot w_2 + \dots + z_{1n} \cdot w_n \quad ,$$

$$A_j' = \sum_{i=1}^n z_{ji} \cdot w_i, \quad j \in [1, m] \quad ,$$

т.е. результирующий показатель – это сумма всех значений, по каждому критерию умноженная на его вес. Веса критериев были расставлены таким образом, чтобы сумма всех весов составляла 1.

7. Упорядочить алгоритмы

Алгоритм, чей результирующий показатель оказался самым большим, является наиболее хорошим для СТЗ. Однако последующие этапы тестирования и адаптации могут выявить недостатки при его работе, в связи с этим все алгоритмы следует упорядочить по убыванию результирующего показателя и при необходимости протестировать второй или третий алгоритм.

Третий шаг разработанной технологии отвечает за процесс тестирования.

Выбранный алгоритм дает наиболее хорошие показатели при обработке одного изображения, на котором вычислялись значения по критериям всех алгоритмов, и даже если это изображение было близко к тем, с которыми алгоритм будет работать в реальной системе, по одному изображению невозможно утверждать о стабильности работы алгоритма. В связи с этим этап тестирования включает в себя прогонку алгоритма по последовательности изображений или видеоряда.

Для примера опишем механизм оценивания по 1 критерию. При тестировании определяется разница между фактическим временем и контрольным, и эта разница сравнивается с установленным пределом: $t_i^{фп} - t_i^{кп} = \Delta t_i^{\Pi} \leq \Delta t^{\Pi}$.

Расчет общей задержки обработки всего видеоряда: $\Delta T^{\Pi} = \sum_{i=1}^{I_{\Pi}} (\Delta t_i^{\Pi} > \Delta t)$, $i = 1, I_{\Pi}$.

Среднее время задержки: $\Delta t_{ср}^{\Pi} = \frac{\Delta T^{\Pi}}{I_{\Pi}}$.

Находим математическое ожидание для определения среднего времени обработки всех кадров: $\tilde{m}_x = \frac{1}{n} \sum_{i=1}^n x_i$.

Рассчитываем дисперсию: $\tilde{D}_x = \frac{1}{n-1} \sum_{i=1}^n (x_i - \tilde{m}_i)^2$.

Среднеквадратичное отклонение показывает, сколько кадров при обработке вышли за рамки допустимых значений: $\tilde{\sigma}_x = \sqrt{\tilde{D}_x}$.

По окончании тестирования выдается отчет, в котором содержится график, представленный на рис.3.

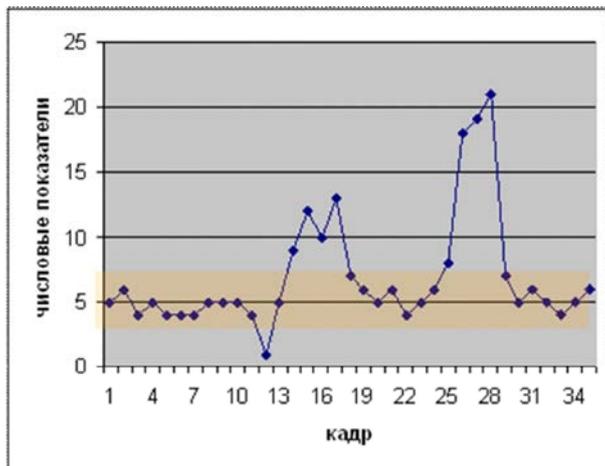


Рис.3. График показателей времени по каждому кадру

По оси x – номер кадра; по оси y – показатели по критерию (время), также отмечен допустимый предел разброса значений.

Визуальный анализ этого графика позволит определить, при обработке каких кадров алгоритм не справился и превысил допустимое время обработки. Более детальный анализ каждого плохо обработанного кадра отдельно позволит принять решение об изменении алгоритма, замене его другим, либо о пренебрежении данными ошибками.

Четвертый, заключительный шаг разработанной технологии позволяет произвести адаптацию ПО СТЗ к условиям эксплуатации.

Под условиями эксплуатации понимают характеристики вычислительной техники, на которой будет работать ПО.

Определение времени, которое будет занимать полная обработка изображения разработанным алгоритмом на предполагаемой, но недоступной ВТ, будет зависеть от трех параметров.

Первый – это разница времени на ВТ разработчика и ВТ реальной системы. Второй – это соотношение размера реального изображения и эталонного, на котором проводилось тестирование. Третий – количество кадров в секунду, требующих обработки.

При разработке системы было определено время работы тестового алгоритма на тестовой платформе:

$$t_d = f(p_d, a_t),$$

где t_d – время работы тестового алгоритма на аппаратной платформе разработчика САПР (d – developer); p_d – характеристики аппаратной платформы разработчика; a_t – тестовый алгоритм (t – test).

Для определения тестового алгоритма на разных платформах $T_{ns} = f(P_{ns}, a_t)$, где T_{ns} – вектор временных характеристик платформ для обучающей выборки нейронной сети; P_{ns} – вектор характеристик аппаратных платформ, используемых для получения обучающей выборки.

Имея входные и выходные значения, можно приступить к обучению нейронной сети:

$$\text{learn}(NS, P_{ns}, T_{ns}),$$

где learn – функция обучения нейронной сети. Ее параметры: NS – нейронная сеть.

Настроили нейронную сеть. Проектировщик разрабатывает алгоритм и тестирует его на своей платформе:

$$t_{pr} = f(p_{pr}, a),$$

где t_{pr} – время работы спроектированного пользователем САПР алгоритма на аппаратной платформе проектировщика; p_{pr} – характеристики аппаратной платформы проектировщика; a – разработанный алгоритм.

Время работы тестового алгоритма на аппаратной платформе проектировщика:

$$t_{prt} = f(p_{pr}, a_t).$$

Рассчитанное нейронной сетью предполагаемое время работы тестового алгоритма на аппаратной платформе проектировщика: $t'_{prt} = \text{run}(NS, p_{pr})$.

Далее определяем трудоемкость:

$$d = \frac{t_{pr}}{t'_{prt}},$$

d – трудоемкость (сложность) разработанного проектировщиком алгоритма в сравнении с тестовым алгоритмом.

Предполагаемое время работы тестового алгоритма на целевой (target) аппаратной платформе, рассчитанное нейронной сетью: $t'_t = \text{run}(NS, p_t)$.

Время работы спроектированного алгоритма на целевой аппаратной платформе: $t_t = t'_t \cdot d$.

Погрешность определения времени работы тестового алгоритма, вычисляемого нейронной сетью:

$$\Delta' = \frac{(t_{\text{prt}} - t'_{\text{prt}})^2}{t_{\text{prt}}}$$

Погрешность определения времени работы спроектированного алгоритма на целевой аппаратной платформе: $\Delta = \Delta' \cdot t_t$.

Технология реализована с помощью использования нейронной сети. Структурная схема процесса представлена на рис.4.

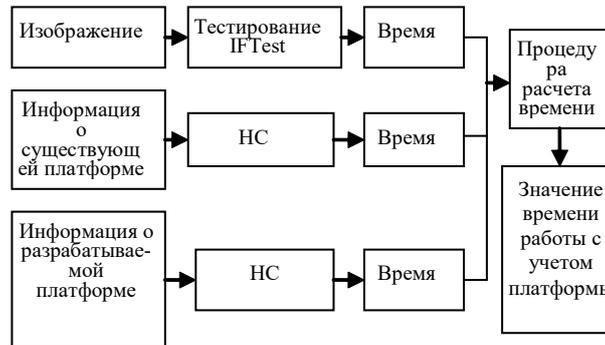


Рис.4. Схема этапа адаптации

Описанный четвертый шаг разработанной технологии позволяет без натурных испытаний определить пригодность разработанного ранее алгоритма обработки изображений в СТЗ на предполагаемой ВТ, что дает возможность увеличить скорость разработки и сократить затраты.

Выводы по данному исследованию и перспективы дальнейших разработок в этом направлении

Предложена технология разработки ПО СТЗ, тестирования и адаптации к условиям эксплуатации.

Описанная технология позволяет за короткое время и с минимальными затратами разработать ПО, т.е. алгоритм обработки изображений для СТЗ, провести тестирование и адаптацию его к условиям эксплуатации. Пошаговая структура технологии позволяет вносить изменения в алгоритм на любом шаге, и в случае отрицательных результатов производить изменения как структуры ПО, так и корректировку внутренних параметров методов (пороги, коэффициенты, ...).

Разработанная технология базируется на:

- продукционно-фреймовой модели для определения этапов и методов обработки изображений и формирования множества алгоритмов;
- несколько измененных аксиоматических методах принятия решения для выбора наиболее хорошего алгоритма из множества сгенерированных алгоритмов;
- нейросетевых технологиях для определения времени работы алгоритма на недоступной ВТ или определения минимальных требований к ВТ.

Список литературы: 1. *Гарячевская И.В., Кузёмин А.Я.* Автоматизация процесса разработки и отладка алгоритмов обработки изображений для СТЗ // Искусственный интеллект. 2004. № 2. С. 269-273. 2. *Гарячевская И.В., Кузёмин А.Я., Кравцов К.А.* Автоматизированная система для проектирования СТЗ мобильного робота // Искусственный интеллект. 2004. № 2. С. 274-278.

Поступила в редколлегию 15.09.2007

Гарячевская Ирина Васильевна, аспирантка, м.н.с. кафедры Информатики ХНУРЭ. Научные интересы: построение СТЗ, обработка изображений. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-14-38, e-mail: irisha@imagefilterer.com.ua.