

**ФАКС НТУУ “КПІ”**

**Лаврущенко О. М.**

**Система візуального керування рухомим  
об'єктом на базі потокової моделі обробки  
даних**

# Вступ

- Розробка візуально-керованих пристроїв є актуальною проблемою
- При цьому для них характерна висока складність [1 , Baroth and Hartsough, 1995].
- Повторне використання програмного коду та вірно обрана архітектурна база можуть значно зменшити складність, а відповідно і час розробки.

# Вступ

- Широке застосування у багатьох інженерних задачах ОС загального призначення Linux.
- Вона вже має характеристики вбудованої ОС з детермінованим часом відгуку.
- Для вирішення завдання візуального керування на цій платформі існують рішення промислового типу, наприклад бібліотека Intel® OpenCV.

# Постановка задачі

- Мета дослідження — створити програмну базу алгоритмів для візуального керування рухомим об'єктом.
- Важливим моментом є спрощення процесу створення алгоритмів та модифікації ланцюгів обробки даних.

# Використання потокової моделі

- Дозволяє максимально розділити між собою алгоритми на функціональному та структурному рівнях, та на рівні вихідного коду.
- Можливе відокремлення конкретної структури системи керування від вихідного коду.
- Можливе також графічне налаштування шаблонів ланцюгів обробки.

# Використання потокової моделі

- Інструменти
- Потокова модель
- Приклади
  - Приклад 1
  - Приклад 2
  - Приклад 3
  - Приклад 4

# Використання потокової моделі Інструменти

- Для Linux є три програмних рішення з поточковим підходом: GStreamer та MLT.
- GStreamer має ширше застосування та підтримку індустрії. Реалізовано на C, за рахунок чого досягається висока швидкість виконання, також можливе використання апаратних ресурсів [2 , Fischer, 2008].

# Використання потокової моделі Інструменти

- Головна ідея - графи фільтрів.
- Вхідні дані проходять чергу обробників «чорних скриньок» вилаштованих у вигляді довільного [6] графу.
- Про скриньки відома лише кількість і властивості їх портів вводу-виводу.
- Елементами можуть бути витoki, стоки, фільтри.



# Використання потокової моделі

- Конкретна послідовність алгоритмів обробки відокремлена від реалізацій окремих алгоритмів
- Граф може бути налаштовано під час виконання, що дає велику гнучкість при роботі з потоками даних у задачах керування.
- З'являється можливість додати системі адаптивні властивості.

# Використання потокової моделі

- Граф обробки даних описується багатопоточною деревоподібною структурою, чи сукупністю дерев.
- Він може описувати складні послідовності з циклами та іншими елементами.
- Дерева можуть виконуватись паралельно, можлива модифікація структури дерева під час виконання.

# Використання потокової моделі

## Приклади

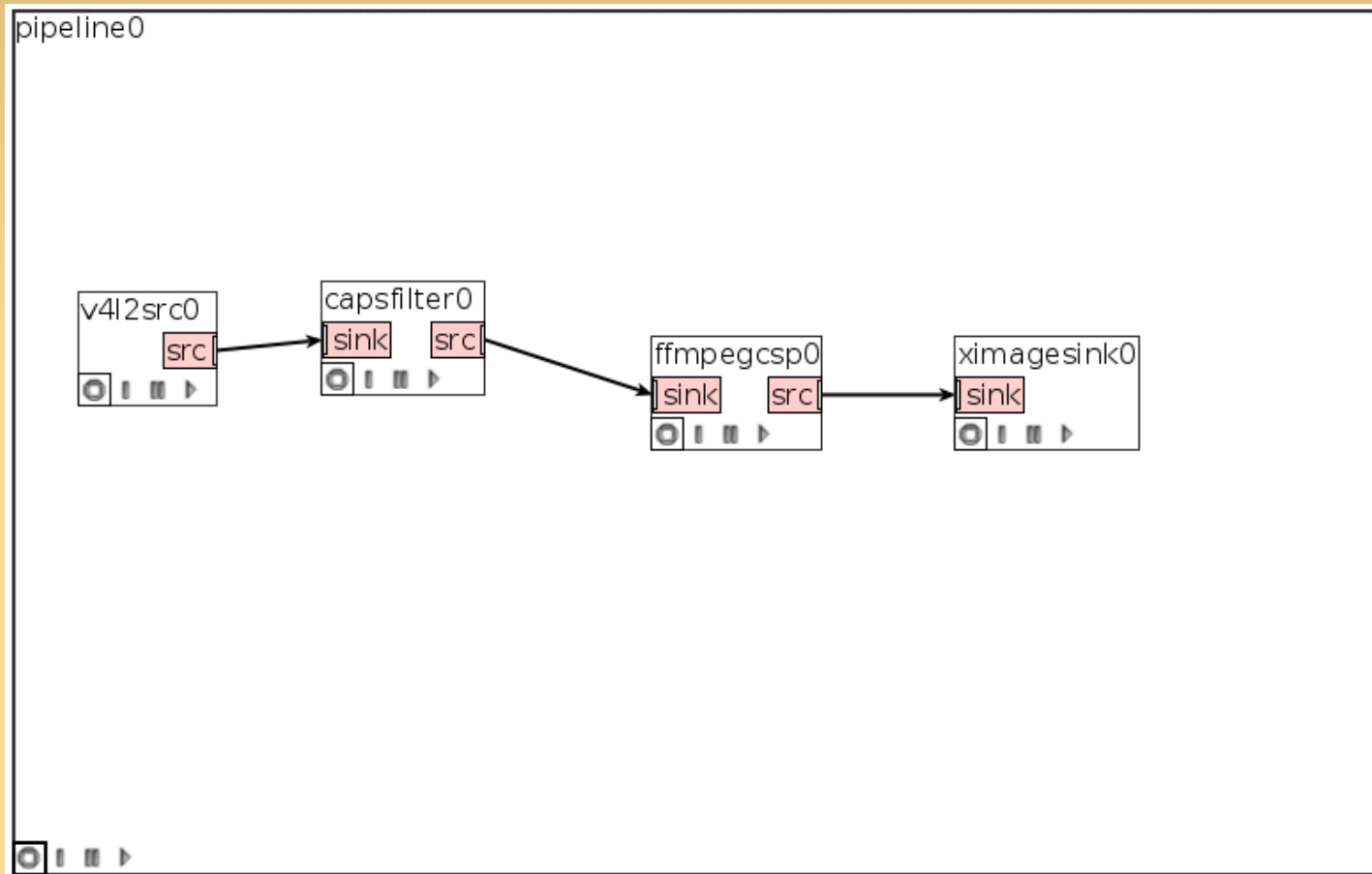


Рис. 1. Редактор графів

# Використання потокової моделі

## Приклад 1

- У якості прикладу дерева можна розглянути завдання передачі відеоданих з бортової камери на наземну станцію моніторингу. Це завдання можна розв'язати різними способами, конкретна конфігурація буде залежати від вимог до якості зображення, пропускної здатності каналу та ін.

# Використання потокової моделі

## Приклад 1

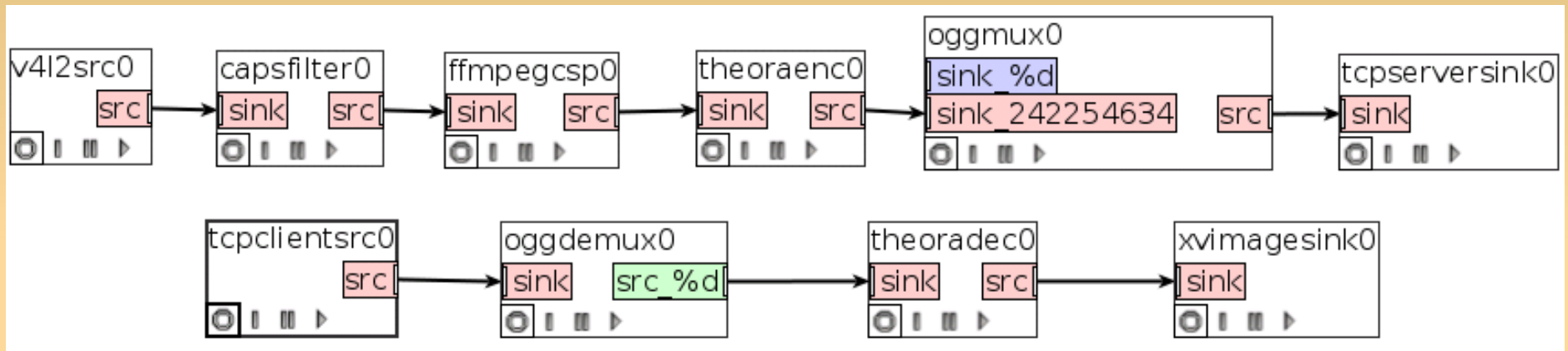


Рис. 2. Модель потоку відеоданих на станцію моніторингу

# Використання потокової моделі

## Приклад 1

- Сирі дані спочатку кодуються згідно вимог, потім стискаються кодеком Theora [7], потім вкладаються у Ogg [8] -контейнер, після чого отримані пакети даних у якості корисного навантаження передаються у TCP-пакетах по мережі.
- На станції моніторингу вони проходять зворотню обробку і результат відображається на ГІК оператора.

# Використання потокової моделі

## Приклад 2

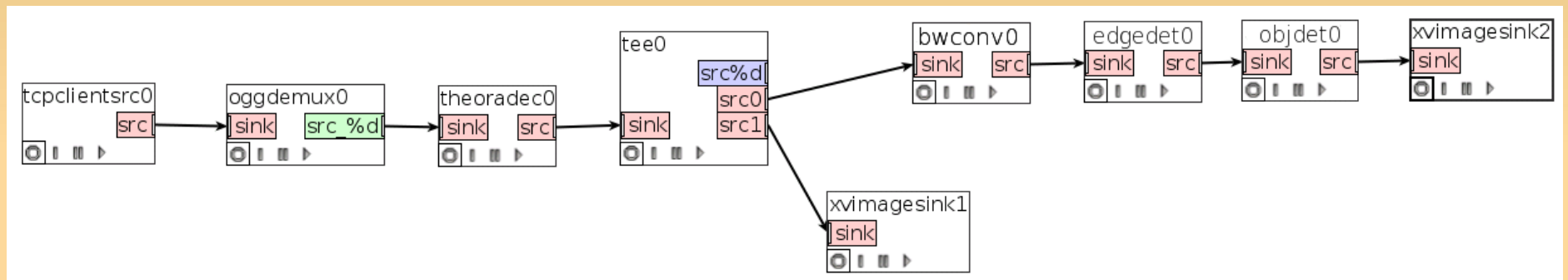


Рис. 3. Варіант моделі обробки на станції моніторингу

# Використання потокової моделі

## Приклад 2

- Структура графу дозволяє оператору спостерігати потік до та після обробки.
- Оператору виводиться відеопотік з камери та окремо виділенна інформація про локалізацію після обробки.
- Фільтри `bwconv`, `edjdet` та `objdet` є власноруч реалізованими алгоритмами: редукції кольорового зображення у бінарне, визначення кордонів об'єктів у кадрі, детектування об'єкту-цілі.



# Використання потокової моделі

## Приклад 3

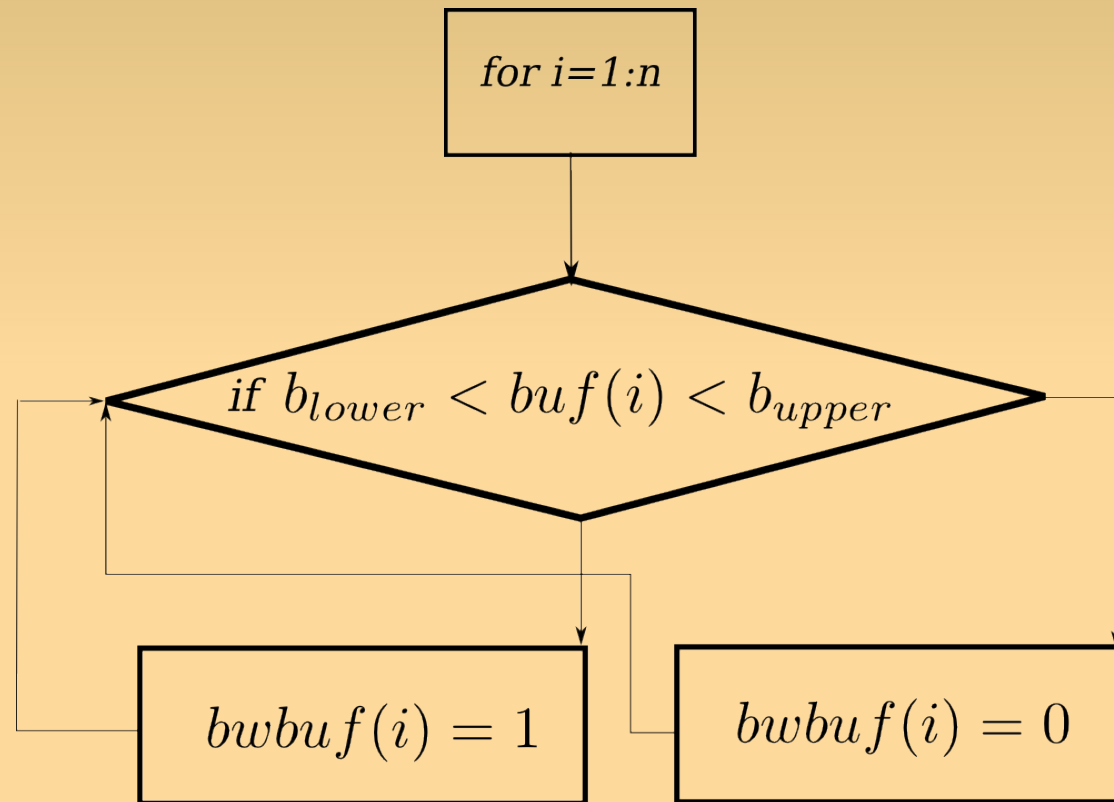


Рис. 3. Цикл редукції кольорів

# Використання потокової моделі

## Приклад 3

- Блок-схема модулю редукції кольорів зображена на рис.3 , де  $\text{buf}[n]$  — вхідний буфер (кадр).  $\text{bwbuf}[n]$  — вихідний буфер. Детально алгоритми показані у [3 ,4 ,5 , Lavrushchenko and Shevchenko].

# Використання потокової моделі

## Приклад 4

- Можливе створення заданих конфігурації.
- Граф може бути створено візуально у графічному редакторі і збережено у вигляді XML-дерева.
- Внутрішнє дерево даних програми також може бути серіалізовано і збережено.

# Використання потокової моделі

## Приклад 4

```
<?xml version="1.0"?>  
<gststreamer xmlns:gst="http://gststreamer.net/gst-core/1.0/">  
  <gst:element>  
    <gst:name>cs udp streaming serv</gst:name>  
    <gst:type>pipeline</gst:type>  
    <gst:param>  
      <gst:name>name</gst:name>  
      <gst:value>cs udp streaming serv</gst:value>  
    </gst:param>  
    <gst:param>  
      <gst:name>async-handling</gst:name>  
      <gst:value>FALSE</gst:value>  
    </gst:param>.
```

# Використання потокової моделі

## Приклад 4

- Можливе створення бібліотеку шаблонів конфігурацій графів обробки даних, що будуть використовуватися при тестуванні або роботі системи керування та станції моніторингу рухомого об'єкту.

# Висновки

- За допомогою комплексу GStreamer реалізовано елементи системи керування рухомим об'єктом, описаної у роботах [3 ,4 ,5 , Lavrushchenko and Shevchenko], елементи-фільтри редукції кольорів, виділення кордонів, пошуку об'єкту.

# Висновки

- Використання Linux та GStreamer у якості серединного шару між апаратним забезпеченням та алгоритмами керування та моніторингу дозволило значно спростити процес розробки та отримати гнучке програмне рішення.

# Бібліографічний список

- [1] E. Baroth, C. Hartsough. Visual programming in the real world // In Visual Object-Oriented Programming: Concepts and Environments. Prentice-Hall, Upper Saddle River, NJ, 21–42, 1995
- [2] T. Fischer. Embedded streaming media with GStreamer // ESC321 Boston, 2008.
- [3] O. M. Lavrushchenko, V. Yu. Shevchenko. Software architecture for control system of visual-based mobile robot// 1st International Scientific Conference «Intelligence, Integration, Reliability», Kiev, 2008.



# Бібліографічний список

[4] О. М. Лаврущенко. Система візуальної орієнтації мобільного роботу під керуванням rtai/debian // VI Всеукраїнська конференція розробників та користувачів вільних програм «OSDN», 2007.

[5] О. М. Лаврущенко В. Ю. Шевченко. Керування рухомим апаратом за візуальною інформацією // Матеріали VI міжнародної науково-технічної конференції «Гіротехнології, навігація, керування рухом і конструювання авіаційно-космічної техніки», присвяченої 100-річчю з дня народження академіка С.П. Корольова, Том №2:150–153, 2007.

# Бібліографічний список

- [6] Документація Gstreamer, <http://gstreamer.freedesktop.org/documentation/>
- [7] Документація Theora, <http://theora.org/>
- [8] Документація Ogg, <http://ru.wikipedia.org/wiki/Ogg>