

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА
АКАДЕМІЯ»

Факультет інформатики

Кафедра інформатики

**Створення додатку для знаходження змін сцен відео в
реальному часі на андроїд
Текстова частина до курсової роботи
за спеціальністю 121 «Інженерія програмного забезпечення»**

Керівник курсової роботи

Бучко О. А.

(прізвище та ініціали)

(підпис)

“ ____ ” _____ 2022 р.

Виконав студент

Іванов А.І.

(прізвище та ініціали)

(підпис)

“ ____ ” _____ 2022 р.

Київ-2022

Тема: Створення додатку для знаходження змін сцен відео в реальному часі на андроїд

Календарний план виконання роботи:

№	Назва етапу курсової роботи	Термін виконання	Примітка
1.	Отримання завдання на курсову роботу	листопад	
2.	Огляд літератури за темою роботи	грудень	
3.	Створення практичної частини роботи	січень-березень	
4.	Написання текстової частини роботи	квітень	
6.	Надання роботи керівнику для перевірки	кінець квітня	
8.	Коригування роботи за результатами перевірки	кінець квітня	
9	Подання роботи на кафедру для перевірки на плагіат	тиждень до захисту	
10.	Захист курсової роботи	кінець травня	

Зміст

1. Вступ.

1.1. Актуальність теми.

1.2. Мета і завдання дослідження.

2. Теоретичні основи:

2.1. Принцип роботи алгоритмів background subtraction.

2.1.1. KNN

2.1.2. MOG2

2.1.3. GMG

2.2. OpenCV: Основні функції, можливості і застосування.

2.3. Виявлення змін сцен у відео: аналіз існуючих методів та підходів.

3. Технології та інструменти.

3.1. Архітектура платформи Android.

3.2. Мова програмування та SDK.

3.3. OpenCV для Android: процес встановлення та інтеграції.

4. Розробка додатку.

4.1. Архітектура додатку.

4.2. Розробка алгоритму виявлення змін сцен.

4.3. Розробка користувацького інтерфейсу.

5. Тестування програми.

6. Висновок.

7. Список використаних джерел.

ВСТУП

1.1. Актуальність теми.

Актуальність теми знаходження змін сцен відео в реальному часі на андроїд обумовлена рядом факторів.

По-перше, з ростом кількості відео контенту в інтернеті, набуває актуальності обробка та аналіз відео. Особливо це стосується областей, де важливо швидко знаходити і відокремлювати важливі моменти в відео, наприклад, відеонагляд, відеоаналітика, розробка редакторів відео, обробка відеоматеріалів для соціальних мереж та ін.

Також, застосування мобільних технологій в аналізі відео дозволяє робити це в реальному часі, незалежно від місця перебування користувача. Це особливо важливо в умовах постійного росту мобільності людей і їх залежності від мобільних пристроїв.

Крім цього, використання алгоритмів background subtraction і бібліотеки OpenCV дозволяє створювати ефективні рішення для аналізу відео, що є актуальним в контексті потреби оптимізації обчислювальних ресурсів мобільних пристроїв.

Тому, створення Android-додатку для виявлення змін сцен у відео в реальному часі є актуальним і перспективним напрямком, що може мати важливе практичне застосування.

1.2. Мета і завдання дослідження.

Мета дослідження полягає в розробці Android-додатку, який дозволить виявляти зміни сцен в відео в реальному часі за допомогою алгоритмів background subtraction, реалізованих з використанням бібліотеки OpenCV.

Для досягнення цієї мети перед нами поставлені такі завдання:

1. Дослідити теоретичні основи виявлення змін сцен в відео, вивчити принципи роботи алгоритмів background subtraction та бібліотеку OpenCV.
2. Проаналізувати можливості платформи Android для реалізації відеоаналізу в реальному часі, вивчити мову програмування та основні елементи Android SDK для розробки додатків.
3. Розробити архітектуру додатку, опираючись на принципи модульності, визначити основні класи та їх функціональність.
4. Реалізувати алгоритм виявлення змін сцен, що включає аналіз відео кадр за кадром, застосування алгоритмів background subtraction для аналізу кадрів та збереження зображень кадрів, коли виявляється зміна сцени.
5. Розробити користувацький інтерфейс, що включає дизайн екранів та елементів управління додатку та їх реалізацію.
6. Провести тестування додатку та оцінити його ефективність за допомогою методів тестування та аналізу результатів.
7. Проаналізувати результати дослідження, визначити обмеження та можливості подальшого покращення та розвитку додатку.

Теоретичні основи

2.1. Принцип роботи алгоритмів background subtraction.

Алгоритми background subtraction, відомі також як віднімання тла, використовуються для відділення першого плану від тла на відео. Ці алгоритми використовуються в багатьох застосуваннях, включаючи відеоспостереження, аналіз руху і, звичайно, виявлення змін сцен.

Загалом принцип роботи алгоритмів background subtraction полягає в наступному:

- Створення моделі тла (background model): На початку відеоряду або протягом певного періоду часу створюється модель тла, яка відображає статичні об'єкти сцени. Ця модель може бути просто середнім значенням кожного пікселя протягом певного часу, або більш складною моделлю, яка враховує зміни освітлення, тіні тощо.
- Віднімання тла (background subtraction): Для кожного нового кадру відео від моделі тла віднімається поточний кадр, що в результаті дає зображення "першого плану", яке містить об'єкти, що відрізняються від тла. Зазвичай, рухомі об'єкти або зміни сцени будуть виділятися на цьому зображенні.
- Оновлення моделі тла: Модель тла може бути постійно оновлювана, щоб враховувати зміни в сцені, що можуть виникнути з часом. Метод оновлення може вар'юватися в залежності від конкретного алгоритму background subtraction.

Різні алгоритми background subtraction, такі як MOG (Mixture of Gaussians), MOG2, KNN, GMG (Gaussian Mixture-based Background/Foreground Segmentation Algorithm) тощо, реалізують ці кроки по-різному і мають різні переваги та недоліки в залежності від

конкретних умов відео (освітлення, тіні, шум, динаміка зміни сцени тощо).

2.1.1. Принцип роботи алгоритма KNN.

K-Nearest Neighbors (KNN) - це алгоритм машинного навчання, що використовується для задач класифікації та регресії. У контексті віднімання тла, алгоритм KNN допомагає відрізнити об'єкти першого плану від тла, створюючи модель тла для кожного пікселя.

Алгоритм KNN віднімання тла працює за наступним принципом:

1. Створення моделі тла: Для кожного пікселя відео алгоритм KNN створює модель тла, яка складається з K найближчих в часі значень пікселів. Кожне значення пікселя включає колір та час входження, і вони сортуються за відстанню до поточного значення пікселя.
2. Віднімання тла: Коли приходить новий кадр, для кожного пікселя визначається, чи відноситься він до тла чи до об'єкта першого плану. Це визначається на основі порівняння поточного значення пікселя з його моделлю тла. Якщо поточне значення пікселя відрізняється від значень у моделі тла більше, ніж на встановлену порогову величину, то піксель вважається пікселем першого плану.
3. Оновлення моделі тла: Модель тла постійно оновлюється, щоб відображати поточну сцену. Кожен новий кадр може призвести до додавання

нових значень до моделі тла або видалення старих значень.

2.1.2. Принцип роботи алгоритма MOG2.

MOG2 (Mixture of Gaussians 2) є одним з популярних методів віднімання тла, який використовує суміш Гауссових розподілів для моделювання кожного пікселя на відео. Він був вперше представлений в 2001 році Зораном Зівковичем і вдосконалений у 2004 році.

Принцип роботи MOG2 можна описати наступним чином:

1. Моделювання пікселів: Для кожного пікселя відео створюється модель, яка складається з декількох (зазвичай 3-5) Гауссових розподілів. Кожен Гауссовий розподіл відповідає деякому кольору або інтенсивності пікселя, який може бути спостережений в даній точці.
2. Оновлення моделі: Коли приходить новий кадр, для кожного пікселя визначається, якому Гауссовому розподілу він відповідає. Якщо значення пікселя відповідає одному з існуючих розподілів (в межах певного порогу), то цей розподіл оновлюється. Якщо жоден розподіл не відповідає значенню пікселя, то створюється новий розподіл.

3. Віднімання тла: Для віднімання тла Гауссові розподіли для кожного пікселя сортуються за значенням ваги/дисперсії. Розподіли з найбільшим значенням ваги/дисперсії вважаються тлом. Значення пікселя вважається пікселем першого плану, якщо воно відрізняється від всіх розподілів тла.

Однією з особливостей MOG2 є те, що він може враховувати різні зміни освітлення, тіні та інші динамічні зміни в сцені завдяки використанню декількох Гауссових розподілів для моделювання кожного пікселя. Крім того, він може адаптуватися до змін у сцені за рахунок динамічного оновлення моделей.

2.1.3. Принцип роботи алгоритма GMG.

GMG (Graph Cuts and Gaussian Mixture Models) - це алгоритм віднімання тла, що був представлений Andrew B. Godbehare, Akihiro Matsukawa, і Ken Goldberg у 2012 році. Його основна ідея полягає в тому, щоб використовувати статистичні методи для моделювання кожного пікселя та застосовувати метод графового різання для визначення регіонів переднього та заднього плану.

Принцип роботи GMG можна описати наступним чином:

Ініціалізація: GMG використовує початкову фазу ініціалізації, протягом якої він вивчає статистику кожного пікселя на відео без впливу переднього плану. Зазвичай ця фаза триває кілька секунд або декілька десятків кадрів.

Створення моделі тла: Для кожного пікселя створюється статистична модель на основі гауссової суміші. Ця модель оновлюється з кожним новим кадром, що дозволяє адаптуватися до змін у сцені.

Сегментація: Після створення моделей тла для всіх пікселів, GMG використовує алгоритм графового різання для сегментації кадру на регіони переднього та заднього плану. Графове різання - це потужний алгоритм, що використовує графи для визначення оптимальної сегментації зображення.

Постпроцесинг: Після сегментації, GMG застосовує постпроцесинг, який може включати морфологічні операції для видалення шуму та покращення результату.

Однією з переваг GMG є його здатність до адаптації до змін в сцені та ефективної роботи в різних умовах освітлення та інших сценаріях. Однак, цей алгоритм має певні недоліки, такі як відносно велика обчислювальна складність та потреба в початковій фазі ініціалізації, що може уповільнити процес аналізу відео.

2.2. OpenCV: Основні функції, можливості і застосування.

OpenCV (Open Source Computer Vision Library) - це відкрита бібліотека програмного забезпечення, призначена для обробки зображень та реалізації алгоритмів комп'ютерного зору. Вона була створена для того, щоб забезпечити зручний і ефективний інтерфейс для використання в розробці програмного забезпечення, що використовує комп'ютерний зір.

Основні можливості OpenCV включають:

Обробка зображень: OpenCV включає багато алгоритмів для обробки зображень, включаючи фільтрацію, перетворення, сегментацію та виявлення особливостей.

Комп'ютерний зір: OpenCV включає алгоритми для розпізнавання об'єктів, відстеження руху, відновлення 3D структури, виявлення обличчя, визначення постави та інші.

Машинне навчання: OpenCV включає декілька алгоритмів машинного навчання, включаючи нейронні мережі, опорні вектори машин та K-найближчих сусідів.

Відеоаналіз: OpenCV включає алгоритми для аналізу відео, включаючи віднімання тла, відстеження об'єктів і виявлення змін сцен.

Інтерфейси для різних мов програмування та платформ: OpenCV можна використовувати з C++, Python, Java та іншими мовами програмування. Існують версії OpenCV для роботи на платформах Windows, Linux, MacOS, iOS та Android.

В курсовій роботі використано для реалізації алгоритмів віднімання тла, які використовуються для виявлення змін сцен. Крім того, OpenCV може бути використано для обробки відео, що включає зчитування кадрів відео, їх аналіз та збереження результатів.

2.3. Виявлення змін сцен у відео: аналіз існуючих методів та підходів.

Виявлення змін сцен у відео - це важлива область досліджень в комп'ютерному зору, яка має багато застосувань, включаючи відеоаналіз, відеоіндексацію, відеонагляд і т.д. Вона полягає в ідентифікації моментів в відео, коли відбувається зміна сцени - зміна контенту зображення, яка є достатньо великою, щоб вважатися переходом від однієї сцени до іншої.

Основні методи виявлення змін сцен можна розділити на кілька категорій:

Методи на основі гістограм: ці методи порівнюють гістограми кольорів між послідовними кадрами. Якщо різниця в гістограмах перевищує певний поріг, це вважається зміною сцени.

Методи на основі блоків: ці методи ділять кадр на блоки та порівнюють блоки між послідовними кадрами.

Методи на основі віднімання тла, які використовувались для цієї курсової роботи: ці методи створюють модель тла для кожного кадру та визначають зміни сцен, порівнюючи нові кадри з моделлю тла. Методи віднімання тла включають статичні методи (як-от метод середнього значення), а також адаптивні методи, які змінюють модель тла відповідно до змін у відео.

Машинне навчання: З використанням методів машинного навчання, можна навчити модель розпізнавати зміни сцен, використовуючи великі набори даних для тренування.

Технології та інструменти

3.1. Архітектура платформи Android.

Архітектура Android складається з кількох рівнів, починаючи з ядра Linux, яке забезпечує сервіси ядра, такі як драйвери пристроїв, управління пам'яттю та процесами. На наступному рівні знаходиться проміжне програмне забезпечення, яке включає набори служб Android Runtime (ART) і бібліотеки C/C++, що надають різноманітні служби та API для розробки додатків. ART включає в себе віртуальну машину, що виконує байт-код, згенерований компілятором Java.

Потім йде рівень Application Framework, що включає набір служб, які безпосередньо використовуються додатками, такими як менеджер вікон, менеджер вмісту та менеджер місцезнаходження. Все це контролюється системою, що базується на інтентів, що дозволяє додаткам взаємодіяти один з одним і з компонентами системи.

Останнім рівнем є самі додатки, що включає як системні додатки, так і ті, що були встановлені користувачами. Ці додатки використовують API, що надається рівнем Application Framework, та мають доступ до вбудованих служб та функцій платформи Android.

3.2. Мова програмування та SDK.

Kotlin — це статично типізована мова програмування, що була розроблена компанією JetBrains. Kotlin використовується для розробки Android-додатків і офіційно підтримується Google. Мова є сумісною з Java, що дозволяє їй використовувати всю екосистему Java, включаючи бібліотеки і фреймворки. Kotlin включає в себе сучасні функції, які спрощують написання коду і роблять його більш безпечним, зокрема, нульові перевірки, корутини, процедури розширення, імутабельність за замовчуванням та багато іншого.

Android SDK, або Software Development Kit, це набір інструментів, що надаються Google для розробки додатків для платформи Android. SDK включає в себе бібліотеки коду, документацію API, емулятор для тестування додатків, інструменти для налагодження та інше. Бібліотеки коду включають в себе набір класів, які реалізують різні функції Android, такі як взаємодія з апаратним забезпеченням, управління інтерфейсом користувача, використання мережевих служб і багато іншого. SDK також містить інструменти для побудови, тестування і налагодження додатків, які включають в себе Android Studio, офіційне середовище розробки (IDE) для Android.

3.3. OpenCV для Android: процес встановлення та інтеграції.

Інтеграція OpenCV з Android вимагає декількох кроків:

1. Завантаження OpenCV Android SDK: Першим кроком є завантаження Android SDK для OpenCV з офіційного веб-сайту OpenCV. Після завантаження SDK потрібно розпакувати архів в зручне для вас місце.
2. Створення нового проекту Android Studio: Далі створюється новий проект в Android Studio. Важливо вибрати підходящу мінімальну версію SDK для вашого додатку.
3. Імпорт модуля OpenCV в проект Android: У Android Studio виберіть File -> New -> Import Module. У вікні, що відкриється, вкажіть шлях до розпакованого SDK OpenCV (sdk/java), тоді виберіть ОК. OpenCV тепер повинен бути в списку модулів вашого проекту.
4. Додавання залежностей модуля OpenCV: Далі необхідно додати залежність модуля OpenCV до основного модуля вашого додатку. Ви можете зробити це, відкривши файл build.gradle вашого основного модуля і додавши OpenCV до списку залежностей (dependencies).
5. Налаштування бібліотеки native: Щоб використовувати OpenCV в Android, необхідно налаштувати бібліотеку

native. Це включає в себе додавання шляху до бібліотеки native в файл build.gradle.

6. Ініціалізація OpenCV в коді додатку: Перед використанням OpenCV в коді додатку потрібно його ініціалізувати. Це зазвичай робиться в методі onCreate активності або застосунку.

Ці кроки інтегрують OpenCV в Android Studio.

Розробка додатку

4.1. Архітектура додатку.

MainActivity

Головна активність починається з завантаження бібліотеки OpenCV і перевірки необхідних дозволів на доступ до камери та пам'яті. Якщо дозволи не надані, він їх запитує.

Основні компоненти:

SurfaceView: Тут відображатиметься попередній перегляд камери.

ImageViews: Використовуються для відображення оброблених зображень.

Camera: Об'єкт, що представляє камеру пристрою.

BackgroundSubtractorMOG2: Це клас OpenCV, який використовується для віднімання тла в відео або зображеннях.

ArrayList<Mat>: Цей список зберігає кадри для обробки.

Додаток використовує зворотний виклик до поверхневого тримача SurfaceView для керування операціями камери. Коли поверхня створена, камера відкривається, встановлюється її орієнтація, і починається попередній перегляд камери. setPreviewCallback камери використовується для захоплення кадрів з камери та їх обробки.

ImageChecking

Ця активність читає всі зображення, збережені в директорії зображень додатку, вносить їх URI в список та відображає зображення в RecyclerView.

Основні компоненти:

RecyclerView: Тут відображаються зображення з пам'яті.

ImageAdapter: Це спеціалізований RecyclerView.Adapter, який керує завантаженням і відображенням зображень у RecyclerView.

ImageAdapter

Цей клас - адаптер RecyclerView, який керує відображенням зображень в активності ImageChecking. Кожне зображення завантажується з його URI і відображається в ImageView.

ImageModel

Цей клас даних містить одну властивість: Uri, який представляє місце зображення.

4.2. Розробка алгоритму виявлення змін сцен.

Центральним компонентом розробленого додатку є алгоритм виявлення змін сцен. Задача виявлення змін на сцені полягає у виявленні візуальних відмінностей між кадрами відео. На вхід алгоритму подається послідовність кадрів, а результатом його роботи є індикація того, чи відбулася зміна сцени.

1. Використання BackgroundSubtractorMOG2

У цьому додатку для виявлення змін на сцені використовується метод віднімання тла, заснований на міксіу гауссових розподілів (MOG2), який надається бібліотекою OpenCV. Цей метод відрізняється тим, що він може краще впоратися з різними освітленнями, тінями тощо.

2. Процес віднімання тла

Алгоритм працює, обробляючи кожен новий кадр відео та віднімаючи від нього "фоновий" кадр. Результатом є "маска" змін, де білі пікселі вказують на області змін, а чорні - на статичні області.

3. Аналіз змін

Після отримання маски змін, додаток аналізує її, щоб визначити, чи відбулася зміна сцени. Це робиться шляхом порівняння кількості білих пікселів в масці (які вказують на зміни) з певним порогом. Якщо кількість білих пікселів перевищує порівняльний поріг, то вважається, що відбулася зміна сцени.

4. Збереження результатів

Якщо відбулася зміна сцени, додаток зберігає поточний кадр в пам'яті пристрою. Це дозволяє користувачам пізніше переглядати точки змін сцен та перевіряти ефективність алгоритму.

Таким чином, цей алгоритм виявлення змін сцен створює основу для додатку, дозволяючи виявляти зміни в реальному часі і забезпечувати ефективну роботу на мобільних пристроях на базі Android.

4.3. Розробка користувацького інтерфейсу.

Для зручності користувачів було розроблено простий та інтуїтивно зрозумілий користувацький інтерфейс, що дає змогу використовувати можливості нашого додатку максимально ефективно.

1. Головний екран

Після запуску додатку користувач потрапляє на головний екран, де він може почати нову сесію знаходження змін сцен відео. На цьому екрані є декілька основних екранів, один з них відображає поточну інформацію з камери, два екрани знизу, які в два рази менше, відображають маски, перша фотографія відображає фотографію тла з яким порівнювався кадр, а наступна відображає кадр який порівнювався, далі йде фінальне фото яке зберігається пройшло перевірку, та зберігається в пам'яті телефону.

2. Відображення результатів

Для того щоб перейти до відображення результатів, нам потрібно натиснути кнопку «Назад», далі відкриється Переглядач фото який відображає всі картинки які були збережені

Таким чином, розроблений користувацький інтерфейс дозволяє користувачам ефективно використовувати додаток, незалежно від їхнього досвіду роботи з подібними технологіями.

Тестування програми

В ході тестування програми було виявлено ряд помилок та проблем.



Рисунок 1

Першою і найбільш очевидною проблемою було незручне відображення зображень в інтерфейсі користувача. Ця проблема була усунута шляхом внесення відповідних змін до коду програми.

Другою поміченою проблемою було повільне прокручування сторінки перегляду фотографій(Рис 2). При великій кількості завантажених зображень, активність (activity) зазнавала затримок у відповіді, що робило неможливим ефективне використання програми. Для вирішення цієї проблеми було застосовано багатопоточність, що дозволило

прискорити відгук програми та покращити її продуктивність.

Втім, найскладнішою для вирішення виявилась проблема з неправильною колірною гаммою зображень. Проблема полягала в тому, що при перетворенні зображення до формату Mat(), колірна гамма зображення втрачала свою якість. Ця проблема була вирішена шляхом заміни колірного простору на інший, що краще підходить для даного варіанту програми.

В результаті проведених оптимізацій та корекцій, програма стала більш стабільною, швидкою та зручною для кінцевого користувача.



Рисунок 2



Рисунок 3

Висновок

У ході виконання даної курсової роботи було проведено глибоке дослідження існуючих алгоритмів виявлення змін сцен у відео, зокрема методів віднімання фону, таких як KNN, MOG2 та GMG. Ці алгоритми було проаналізовано з точки зору їхньої ефективності, швидкості та придатності для використання в реальному часі на платформі Android.

Було визначено, що мова програмування Kotlin та Android SDK надають потужні інструменти для створення високопродуктивних додатків, і вони були обрані для розробки додатку. Було також встановлено, що OpenCV надає велику кількість функцій для обробки зображень та відео, які можуть бути використані для імплементації алгоритмів виявлення змін сцен.

Для тестування додатку було використано набір відео з різними параметрами, що дозволило оцінити ефективність та точність алгоритмів. Результати тестування показали, що розроблений додаток ефективно виявляє зміни сцен у відео в реальному часі.

Таким чином, створення додатку для виявлення змін сцен у відео в реальному часі на платформі Android є актуальним та перспективним напрямком, який має велике значення для багатьох галузей, включаючи кіноіндустрію, відеонагляд, спорт та інші. Дана робота демонструє можливість імплементації цієї технології в реальному додатку, що може бути використаний на практиці.

Список використаних джерел

Bradski, G., & Kaehler, A. (2008). "Learning OpenCV: Computer vision with the OpenCV library". O'Reilly Media, Inc.

Chum, O., Philbin, J., Sivic, J., Isard, M., & Zisserman, A. (2007). "Object retrieval with large vocabularies and fast spatial matching". In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-8). IEEE.

Zivkovic, Z. (2004). "Improved adaptive Gaussian mixture model for background subtraction". In Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. (Vol. 2, pp. 28-31). IEEE.

Godbehere, A. B., Matsukawa, A., & Goldberg, K. (2012). "Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation". In American Control Conference (ACC), 2012 (pp. 4305-4312). IEEE.

Kotlin documentation. (2023). JetBrains. Retrieved from <https://kotlinlang.org/docs/home.html>

Android Developers. (2023). "Develop apps". Android Developers. Retrieved from <https://developer.android.com/guide>

Lienhart, R., & Maydt, J. (2002). "An extended set of Haar-like features for rapid object detection". In Proceedings. International Conference on Image Processing (Vol. 1, pp. I-I). IEEE.

Dalal, N., & Triggs, B. (2005, June). "Histograms of oriented gradients for human detection". In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) (Vol. 1, pp. 886-893). IEEE.

Viola, P., & Jones, M. (2001). "Rapid object detection using a boosted cascade of simple features". In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001 (Vol. 1, p. I-I). IEEE.