



INF 2102 – Projeto Final de Programação

Detecção, rastreamento e extração de
propriedades de ondas através de visão
computacional

Aluno: Lauro Henriko Garcia Alves de Souza
Matrícula: 1912713

Orientador: Waldemar Celes

Conteúdo

1 Introdução

1.1 Especificação de requisitos

2 Tarefas de implementação para execução do projeto

3 Metodologia

4 Tecnologias

5 Documentação

6 Protótipo

7 Resultados

1. Introdução

Este trabalho tem como objetivo a especificação, o projeto, a implementação e a validação dos resultados de uma aplicação de visão computacional, na qual pode-se extrair propriedades das ondas na zona de surfe através de sensor remoto de vídeo. Esta tarefa será realizada através da combinação dos seguintes métodos: *edge detection* para extração de *feature* e *clusterização* para tracking das ondas.

O objetivo dessa aplicação é fornecer um melhor entendimento sobre o processo de transformação de ondas na zona de surfe e com isso contribuir com trabalho de engenharia costeira e segurança no mar. O resultado deve ser o fornecimento das seguintes informações: quantidade, direção, percurso, localização, duração e frequência das ondas.

O primeiro componente desse sistema é o algoritmo de detecção. Ele começa identificando bordas acima de um *threshold* inicial. Caso haja essa identificação, essa borda é submetida a rotina de *clusterizacao* que agrupa essa borda a uma onda já existente ou marca o início de uma nova onda. Depois da inicialização o *threshold* é submetido a rotina de aprendizado automático e redefinido.

Para a etapa de identificação de bordas, é sobreposto um grid dual na imagem e é feita uma convolução nos boxes do grid para identificar boxes adjacentes na horizontal que tenha a diferença entre eles maior que o *threshold*.

Para a etapa da clusterização, as seguintes dimensões de similaridade são analisadas. X é a coordenada em pixels do eixo x, Y

é a coordenada em pixels do eixo y e t é o momento em que a onda é detectada.

E por fim, existe um processo de raciocínio automático que define o threshold para fazer a detecção do edge.

1.1 Especificação de requisitos

REQUISITOS	
Requisito	Descrição
RF1	Diariamente, entre 6h e 16h, o programa deve detectar e rastrear as ondas durante 50 minutos da hora.
RF2	O programa deve checar a qualidade da transmissão da câmera.
RF3	Ao fim desses 50 minutos, o programa deve salvar no banco de dados todas as ondas detectadas com a sua posicao e horario de inicio, percurso, e sua posicao e horario de fim.
RF4	A cada hora do dia, entre 6:51h e 16:51h, o programa deve criar as seguintes estaticas a partir das ondas inseridas no banco de dados: quantidade de ondas (esquerdas e direitas) com percurso maior que 20 metros, intervalo em minutos entre as séries de ondas, quantidade de ondas a cada 10 minutos, distância média percorrida pelas ondas, duração média em segundos das ondas e a distância da area até a arrebentação.
RF5	O programa deve fornecer uma aplicacao REST para o envio das estatísticas das ondas.
RF6	O programa deve exibir as estatísticas das ondas em uma interface web
RF7	O programa deve exibir um video com a detecção das ondas que aconteceram na ultima hora na mesma interface aonde se encontram as estatísticas.
RF8	O programa deve fornecer, em uma interface web, uma ferramenta de anotação, na qual serão exibidas ondas detectadas pelo algoritmo e o usuário poderá clicar aonde ele detecta a onda.
RF9	As coordenadas em pixel da onda detetcta pelo usuário deverá ser salvo no banco de dados.
RF10	O usuario só poderá salvar a sua detecção de onda se fornecer o email.
RF11	O usuario só pode escolher até 3 ondas

2 Tarefas de implementação para execução do projeto

A sequência de tarefas necessárias para o projeto foi dividida da seguinte forma:

1. Estudo bibliográfico sobre detecção de onda por sensor remoto de vídeo.
2. Pesquisa e experimentação de diferentes métodos de visão computacional.
3. Aplicação de grid regular na imagem.
4. Implementação do algoritmo de detecção através de *edge detection* por segmentação de gradiente.
5. Implementação do algoritmo de *clustering* para o *tracking* das ondas.
6. Testes visuais.
7. Divisão da imagem em um grid dual não-regular.
8. Implementação do algoritmo de aprendizado automático de *threshold* para o *edge detection*.
9. Testes de visualização.
10. Implementação do algoritmo de extração de propriedades das ondas.
11. Implementação da RESTFul API.
12. Implementação da aplicação Web para visualização de dados.
13. Implementação da aplicação Web para validação dos resultados.

A tabela abaixo contém o cronograma das tarefas executadas:

	Janeiro	Fevereiro	Março	Abril	Maio	Jun	Julho
Tarefa 1	x						
Tarefa 2	x	x					
Tarefa 3		x					
Tarefa 4		x	x	x			
Tarefa 5			x	x	x		
Tarefa 6					x		
Tarefa 7					x		
Tarefa 8					x	x	
Tarefa 9					x	x	
Tarefa 10					x	x	
Tarefa 11						x	x
Tarefa 12						x	x
Tarefa 13							x

3 Metodologia

3.1 Aquisição de dados

Uma câmera de rede de definição 1280x720 pixels foi colocada em frente a praia do arpoador no topo do posto salva-vidas que fica a 7 metros do nível do mar.

3.2 Downsampling

Transformação da imagem 1280x720 em dois grids. O primeiro grid de 1260x396, que por sua vez é dividido em boxes 36x36 pixels. O segundo grid 1260x315, que por sua vez é dividido em boxes 45x45 pixels. Esse *downsampling* é feito para reduzir o *noisy* e custo computacional. O boxes tem tamanhos diferentes para ajustar a escala e cada boxe é a média dos pixels que o compõe.

3.3 Edge detection

É calculada a diferença entre boxes adjacentes na horizontal e identificado como onda, os boxes cuja a diferença ultrapassa o threshold.

3.4 Clustering

É avaliado para cada box identificado como onda, se a sua localização e o momento em que ele foi identificado, faz parte de uma onda já existente ou configura uma nova onda.

3.5 Aprendizado automático de Threshold

As diferentes luzes do dia, fazem com que a diferença máxima entre boxes adjacentes varie. A cada frame lido, a diferença máxima entre boxes adjacentes é anotada. A partir de 30 segundos de vídeo o threshold é definido como uma média do 4º quartil.

4 Tecnologias

4.1 Python

Python é uma linguagem de programação interpretada, orientada a objetos e de alto nível, com semântica dinâmica. Suas estruturas de dados de alto nível, combinadas com digitação dinâmica e ligação dinâmica, o tornam muito atraente para o Rapid Application Development.

4.2 Opencv

OpenCV (Open Source Computer Vision Library) é uma biblioteca de software de visão computacional e aprendizado de máquina de código aberto. O OpenCV foi desenvolvido para fornecer uma infraestrutura comum para aplicativos de visão computacional e acelerar o uso da percepção da máquina nos produtos comerciais.

4.3 Numpy

O NumPy é o pacote fundamental necessário para a computação científica com Python. O NumPy também pode ser usado como um eficiente recipiente multidimensional de dados genéricos. Tipos de dados arbitrários podem ser definidos. Isso permite que o NumPy se integre de maneira fácil e rápida a uma ampla variedade de bancos de dados.

4.3 SciPy

O SciPy é um ecossistema baseado em Python de software de código aberto para matemática, ciências e engenharia.

4.4 Pandas

O Pandas é uma ferramenta de análise e manipulação de dados de código aberto rápida, poderosa, flexível e fácil de usar, construído sobre a linguagem de programação Python.

4.5 Django

O Django é uma estrutura da Web Python de alto nível que incentiva o desenvolvimento rápido e o design limpo e pragmático. Construído por desenvolvedores experientes, cuida de grande parte dos problemas do desenvolvimento da Web, para que você possa se concentrar em escrever seu aplicativo sem precisar reinventar a roda. É de código aberto e gratuito.

4.6 RESTFUL API

REST é acrônimo para REpresentational State Transfer. É um estilo arquitetônico para sistemas hipermídia distribuídos e foi apresentado pela primeira vez por Roy Fielding em 2000 em sua famosa dissertação.

4.7 Javascript

JavaScript® (frequentemente abreviado como **JS**) é uma linguagem de programação leve, interpretada e orientada a objetos com funções de primeira classe, conhecida como a linguagem de scripting para páginas Web, mas também utilizada em muitos ambientes fora dos navegadores. Ela é uma linguagem de scripting baseada em protótipos, multi-paradigma e dinâmica, suportando os estilos orientado a objetos, imperativo e funcional.

4.8 MySQL

O MySQL é o banco de dados de código aberto mais popular do mundo. Com desempenho, confiabilidade e facilidade de uso comprovados, o MySQL se tornou a principal opção de banco de dados para aplicativos baseados na Web, usados por propriedades da Web de alto perfil, incluindo Facebook, Twitter, YouTube, Yahoo! e muitos mais.

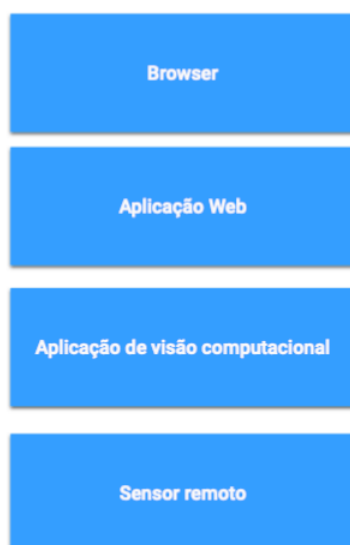
-
- 4.1 <https://www.python.org/doc/essays/blurb/>
 - 4.2 <https://numpy.org/doc/stable/about.html>
 - 4.3 <https://www.scipy.org>
 - 4.4 <https://pandas.pydata.org>
 - 4.5 <https://www.djangoproject.com>
 - 4.6 <https://restfulapi.net>
 - 4.7 https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/About_JavaScript
 - 4.8 <https://www.mysql.com/about/>

5 Documentação

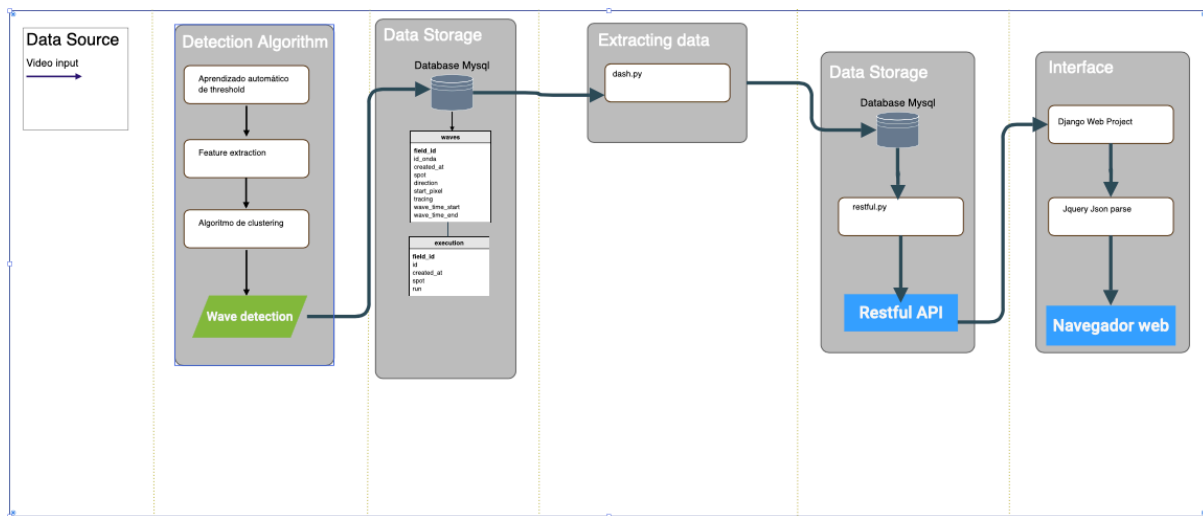
5.1 Descrição geral

Esse sistema foi desenvolvido no paradigma procedural e é composto de 4 subcomponentes: `wave_detection.py`, `dash.py`, `rest_framework` e Django web app. O `wave_detection.py` recebe as imagens da câmera e é responsável pela detecção e rastreamento de ondas, `dash.py` é responsável pela transformação dos dados do `wave_detection.py` em medidas estatísticas. O `rest_framework` transporta essas medidas estatísticas para o Django web app que exibe ao usuário.

5.2 Arquitetura em camadas do projeto:

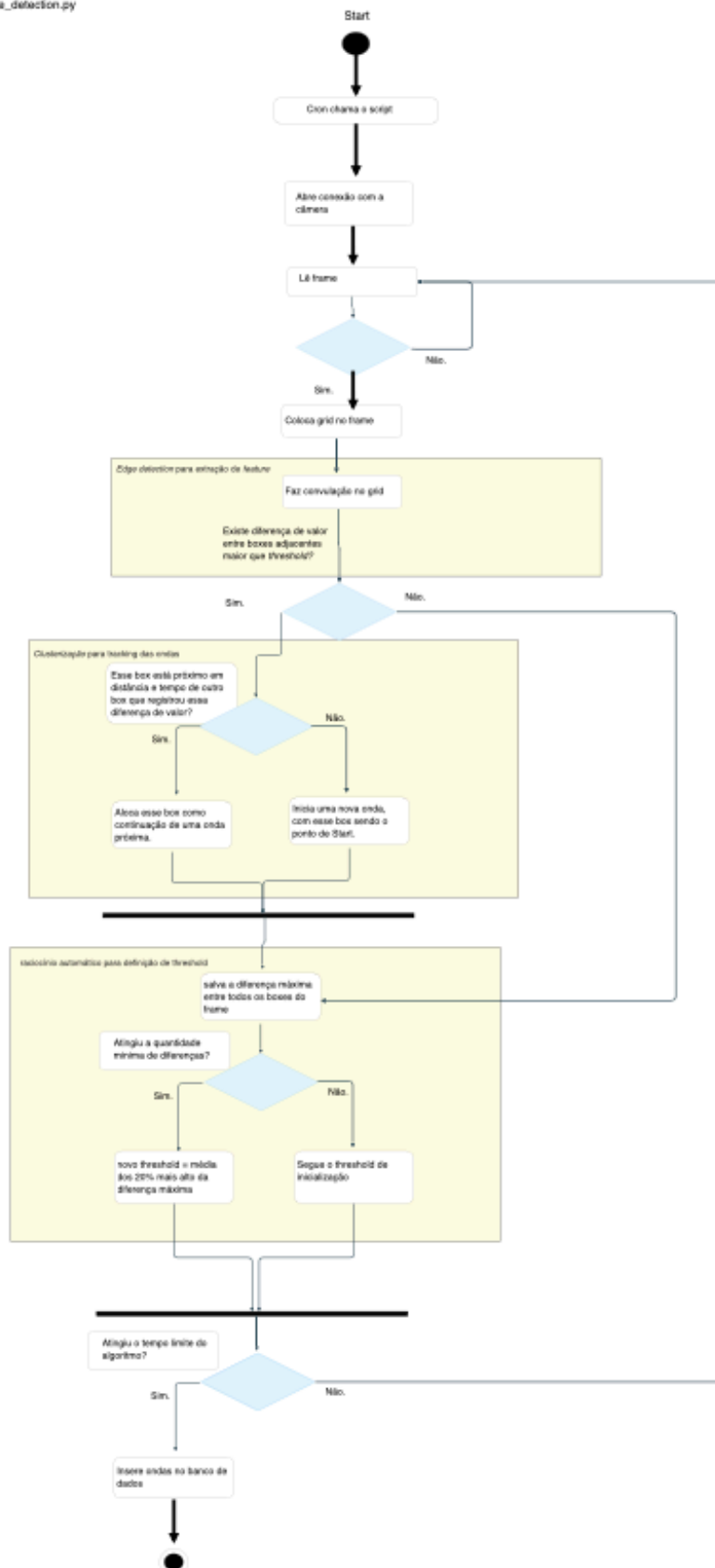


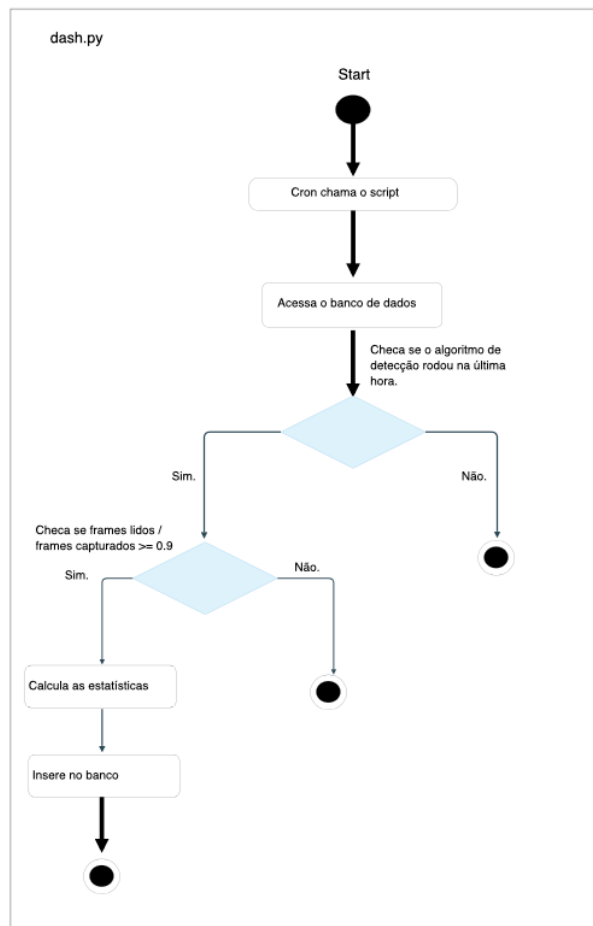
5.3 Arquitetura do projeto com detalhes:



5.4 Diagramas de atividade

wave_detection.py





5.5 Controle de qualidade

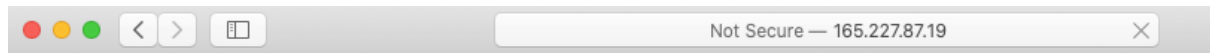
Para controle de qualidade, durante a execução do `wave_detection.py`, é calculado a quantidade de frames lidas e/ou perdidas e quando a taxa de frames perdidos excede 10% do total de frames, as estatísticas não são calculadas.

Nesse caso, fica sendo exibido na interface web, as ultimas estatísticas do dia. Caso no dia, nenhuma rodada do algoritmo tenha ocorrido com a quantidade mínima de frame é exibido um aviso: “Sem dados”.

6 Protótipo

O sistema foi colocado em produção e a visualização das estatísticas em tempo real pode ser acessado através da url:

<http://165.227.87.19/pro/>



INF 2102 - Projeto Final de Programacao - PUC Rio

Aluno: Lauro Souza

Matricula: 1912713

Orientador: Waldemar Celes

Data: **2020-07-01T17:00:03Z**

spot: **arpoador**

waves_longer_20_meters: **19**

left_waves_longer_20_meters: **13**

right_waves_longer_20_meters: **6**

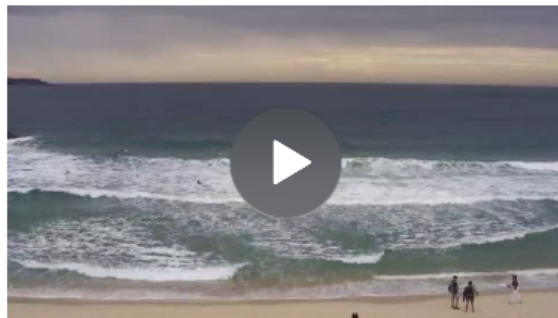
waves_set_interval(minutes): **5**

waves_per_10min(min): **6**

avg_wave_distance(m): **14**

avg_wave_duration(secs): **14**

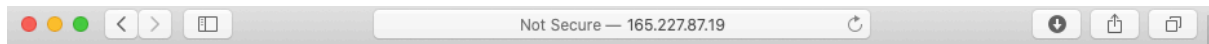
distance_outside(m): **166**



OBS: as linhas desenhadas no video representa o tracking das ondas mais recentes e mais longas

A ferramenta para validação pode ser acessado através da url:

<http://165.227.87.19/gallery/>



INF 2102 - Projeto Final de Programacao - PUC Rio

Aluno: Lauro Souza

Matricula: 1912713

Orientador: Waldemar Celes



Email:

IMPORTANTE

DEFINICAO:
ONDA -> Fronteira entre a espuma e a parede da onda

INSTRUcoes

- Clique em ate 3 ondas
- Clique exatamente na fronteira entre a espuma e a parede da onda
- Caso nao veja essa combinacao de espuma seguido de fronteira, significa que essa imagem esta sem ondas.
- Comece sempre pelas maiores e mais definidas ondas.

Veja o exemplo abaixo:



6 Resultados Visuais

6.1 Detecção

Verificando as primeiras detecções de ondas os últimos 3 dias 01/07, 30/06 e 29/06 nos horários da manhã (08:00), meio do dia (12:00) e a tarde (16:00) (para termos diferentes luzes) podemos ver que detecções foram feitas de forma correta.

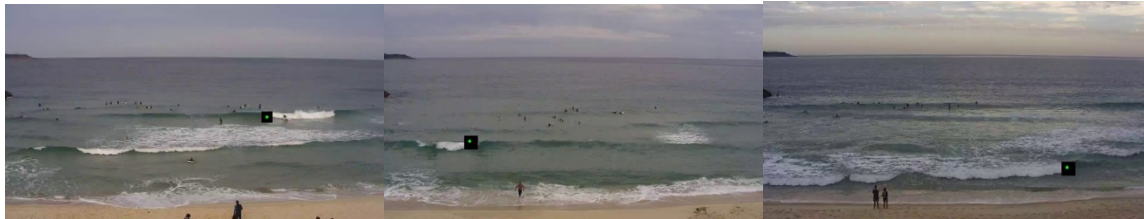
 Detector

Dia 01/06

08:00:06

12:00:06

16:00:51

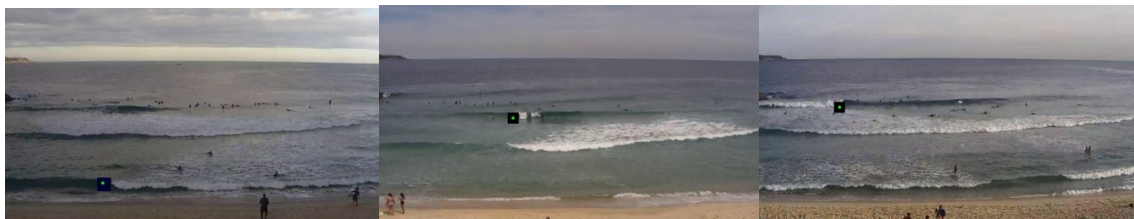


Dia 30/06

08:00:08

12:00:10.

16:00:10

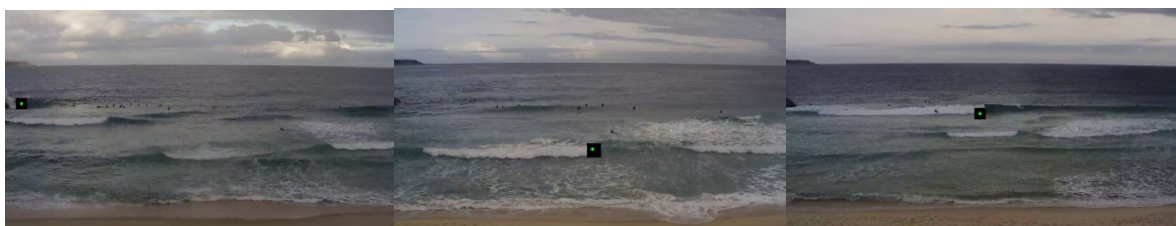


Dia 29/06

08:00:07

12:00:48.

16:00:51



6.2 Rastreamento

Avaliando os vídeos gravados pelo algoritmo no dia 01/07, de manhã, meio do dia e tarde, também é possível ver o rastreamento sendo feito corretamente.

Sequência de frames do vídeo do dia 01/07 as 08:00

— Representa o Tracking da onda



Sequência de frames do vídeo do dia 01/07 as 12:00





Sequência de frames do vídeo do dia 01/07 as 16:00

