

Resolução do Ritual de Iniciação - Implementação SAFT

37 messages

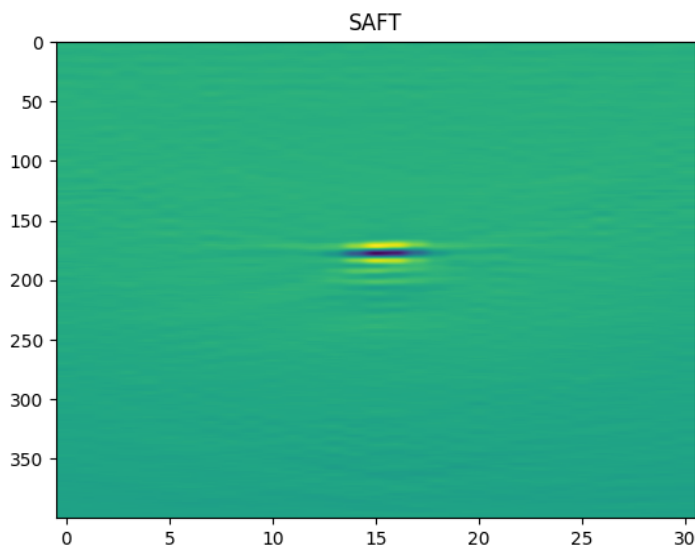
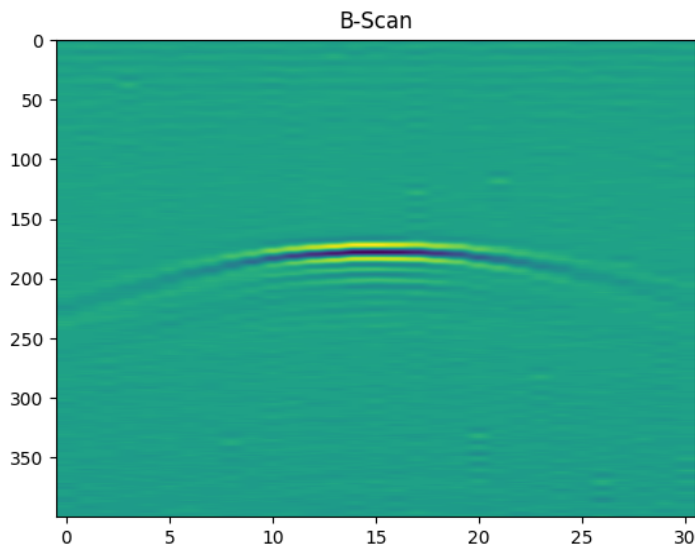
Thiago E. Kalid <thiagokkalid@gmail.com>

17 March 2021 at 17:17

To: "Thiago A. R. Passarin - UTFPR" <passarin@professores.utfpr.edu.br>

Prezado professor Thiago Passarin,

Com a explicação do senhor a ideia do algoritmo ficou claro como o dia em minha cabeça, e consegui já terminar todo o programa. Abaixo consta o resultado:



Segue em anexo o .zip contendo a pasta com meu projeto, assim como apenas o script, caso o senhor tenha interesse de analisá-lo de forma mais rápida e prática.

 [Ritual_de_Iniciacao.zip](#)

 [main.py](#)
5K

Thiago A. R. Passarin - UTFPR <passarin@professores.utfpr.edu.br>
To: "Thiago E. Kalid" <thiagokkalid@gmail.com>

18 March 2021 at 09:41


Legal, Thiago! É esse mesmo o resultado.

Já podemos passar para a próxima etapa, que é o algoritmo TFM. O princípio é o mesmo do SAFT, mas ele tem um conjunto mais completo de dados. Ao invés de um transdutor se deslocando mecanicamente ao longo do objeto, ele utiliza dados de um array de transdutores, onde cada um dispara de uma vez, e a cada disparo todos recebem os ecos. Estou te mandando em anexo o artigo que descreve o TFM. Vou gerar uns dados de simulação para você tentar fazer a reconstrução (espero conseguir fazer isso ainda hoje).

Qualquer dúvida, é só perguntar. Talvez a princípio fique um pouco confusa a parte sobre *analytical signal*, mas é perfeitamente possível executar o algoritmo sem fazer essa transformação nos dados, então não vale a pena travar nesse ponto se estiver muito difícil entender. De qualquer forma, podemos discutir na sequência como funciona essa operação.

Att,
Passarin

[Quoted text hidden]

 **1-s2.0-S0963869505000721-main.pdf**
581K

Thiago A. R. Passarin - UTFPR <passarin@professores.utfpr.edu.br>
To: Daniel Pipa <danielpipa@gmail.com>

18 March 2021 at 09:45

Pipa, o Thiago Kalid respondeu bem. Pretendo colocá-lo para trabalhar nas pendências que foram se acumulando na frente de detecção de superfície.
Vamos adicioná-lo ao grupo de e-mails e do WhatsApp?

Att,
Passarin

[Quoted text hidden]

 **main.py**
5K

Thiago E. Kalid <thiagokkalid@gmail.com>
To: "Thiago A. R. Passarin - UTFPR" <passarin@professores.utfpr.edu.br>

18 March 2021 at 13:24

Prezado professor Passarin,

Vou ler o artigo e me informar melhor. Enquanto isso estou no aguardo dos dados.

Atenciosamente,
Thiago E. Kalid
[Quoted text hidden]

Daniel R. Pipa <danielpipa@gmail.com>
To: "Thiago A. R. Passarin - UTFPR" <passarin@professores.utfpr.edu.br>

18 March 2021 at 16:12

Sim. Eu te promovi a adm da lista de email e grupo de WhatsApp. Pode acrescentá-lo nos dois.

--
Daniel Rodrigues Pipa
Universidade Tecnológica Federal do Paraná

[Quoted text hidden]

Thiago A. R. Passarin - UTFPR <passarin@professores.utfpr.edu.br>
To: "Thiago E. Kalid" <thiagokkalid@gmail.com>

18 March 2021 at 16:58

Olá, Thiago.

Coloquei os dados para o TFM nessa pasta: <https://drive.google.com/drive/folders/1W1NAJIXmbUDHEvAqKjrsWXOiww263b6r?usp=sharing>

Tem um pequeno script ali mostrando como carregar os dados. Tentei deixar os nomes das variáveis auto-explicativas, e com as unidades. Há duas imagens mostrando o cenário simulado e uma imagem com o resultado esperado.

Se você não usar o analytic signal (que faz uma detecção de envoltória complexa), os pontos da sua imagem não vão ficar tão suaves quanto essa que eu coloquei lá. Mas não tem problema.

Espero não ter esquecido nenhuma informação. Pode me contactar se tiver qualquer dúvida.

Att,
Passarin
[Quoted text hidden]

Thiago E. Kalid <thiogokkalid@gmail.com>
To: "Thiago A. R. Passarin - UTFPR" <passarin@professores.utfpr.edu.br>

18 March 2021 at 19:47

Prezado professor Thiago Passarin,

Gostaria de saber como está organizada aquela estrutura de dado chamada ascan, pois percebi que existem 1858 matrizes 64x64 e estou em dúvida de como saber qual a-scan pertence a qual posição. Além disso, a variável: "samples_t_init_microsec" representa o que exatamente? O tempo de início do experimento?

Atenciosamente,
Thiago E. Kalid
[Quoted text hidden]

Thiago A. R. Passarin - UTFPR <passarin@professores.utfpr.edu.br>
To: "Thiago E. Kalid" <thiogokkalid@gmail.com>

18 March 2021 at 20:25

Olá Thiago,

O transdutor possui 64 elementos que trabalham independentemente. Então, é como se cada elemento fosse um transdutor separado. O array chamado ascan tem as seguintes dimensões: amostras x emissor x receptor. Por exemplo, o vetor ascans[:, 10, 15] contém o A-scan (eco) observado pelo elemento 15 quando o elemento 10 disparou.

Seria possível também fazer um SAFT com esses dados. Nesse caso, você usaria apenas os ecos ascans[:, i, i] com i de 0 a 63, ou seja, onde o emissor e o receptor são o mesmo elemento. Aliás, esse é um bom ponto de partida.

A variável samples_t_init_microsec contém o instante de tempo da primeira amostra dos A-scans. Ou seja, das amostras ascans[0, i, j] para qualquer i e j de 0 a 63. Para obter os instantes de tempo das demais amostras, é só somar a este valor múltiplos do intervalo de amostragem (que, por sua vez, é o inverso da frequência de amostragem f_sampling_MHz).

Att,
Passarin
[Quoted text hidden]

Thiago E. Kalid <thiogokkalid@gmail.com>
To: "Thiago A. R. Passarin - UTFPR" <passarin@professores.utfpr.edu.br>

22 March 2021 at 18:39

Prezado professor Thiago Passarin,

Gostaria de esclarecer se eu estou compreendendo ou não o método TFM.

$$I(x, z) = \left| \sum_{\text{for all tx, rx}} h_{\text{tx, rx}} \left(\frac{\sqrt{(x_{\text{tx}} - x)^2 + z^2} + \sqrt{(x_{\text{rx}} - x)^2 + z^2}}{c_1} \right) \right| \quad (12)$$

O $h_{\text{tx}, \text{rx}}$ seria obtido analogamente ao método do SAFT, isto é, o valor Y do $\text{ascan}(Y, i, j)$ que seria equivalente ao tempo necessário para a onda se propagar do transmissor i ao receptor j , para todas as combinações possíveis de transmissor e receptor. Pergunto isso fiz um script que até o momento está tomando um tempo enorme para executar (ainda não consegui terminar de executá-lo), então estou confuso se a demora na execução reside na inerente complexidade computacional do programa ou em uma implementação muito ineficiente de minha parte.

Atenciosamente,
Thiago E. Kalid

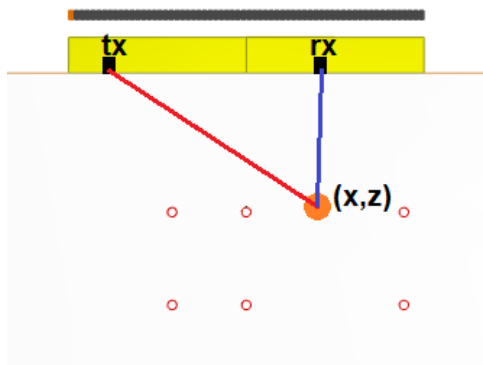
[Quoted text hidden]

Thiago A. R. Passarin - UTFPR <passarin@professores.utfpr.edu.br>
To: "Thiago E. Kalid" <thiagokkalid@gmail.com>

22 March 2021 at 21:21

Olá, Thiago. Vou respondendo abaixo, a partir do seu texto.

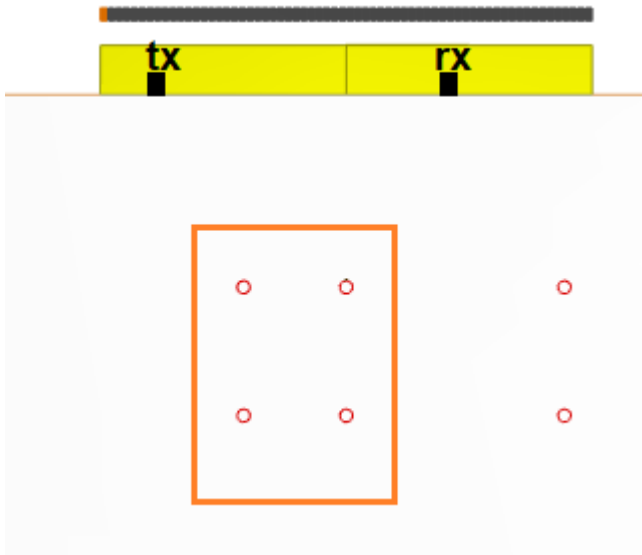
O $h_{\text{tx}, \text{rx}}$ seria obtido analogamente ao método do SAFT, isto é, o valor Y do $\text{ascan}(Y, i, j)$ que seria equivalente ao tempo necessário para a onda se propagar do transmissor i ao receptor j , para todas as combinações possíveis de transmissor e receptor. Sim, é análogo ao SAFT. Mas o tempo total corresponde ao tempo de propagação do transmissor i até o pixel em (x, z) mais o tempo de propagação do pixel em (x, z) até o receptor j . Representei graficamente essas duas parcelas do tempo em vermelho e em azul, respectivamente, no desenho abaixo.



Pergunto isso fiz um script que até o momento está tomando um tempo enorme para executar (ainda não consegui terminar de executá-lo), então estou confuso se a demora na execução reside na inerente complexidade computacional do programa ou em uma implementação muito ineficiente de minha parte.

Sim, a complexidade computacional é muito maior, e provavelmente esse é o motivo do seu script estar levando tanto tempo. No SAFT, para um transdutor de 64 elementos, a quantidade de cálculos de distância é igual ao número de pixels vezes 64. Já no TFM, a quantidade de cálculos de distância é igual ao número de pixels vezes 64^2 , pois existem $64^2 = 4096$ combinações de emissor-receptor. Nós costumamos utilizar funções prontas do scipy que fazem um cálculos de distância otimizados e utilizando paralelismo ([essa](#) especificamente), e também utilizamos o numba, que tem recursos avançados de paralelismo. Para verificar o funcionamento do seu algoritmo sem precisar utilizar essas ferramentas, há duas coisas mais simples que você pode fazer:

1. Reduzir o tamanho da região reconstruída. Você pode, por exemplo, reconstruir a imagem somente em volta de alguns dos furos, como no exemplo abaixo em laranja. Se você reconstruir as coordenadas em x de -11 mm a 3 mm, e as coordenadas em z de 12 mm a 28 mm, você já pega esses quatro furos ali. Isso permite usar uma quantidade menor de pixels.
2. Reduzir a resolução espacial. Essa é outra estratégia para reduzir a quantidade de pixels e, assim, o custo computacional. Isso vai prejudicar a qualidade da imagem, mas por enquanto, como o objetivo é entender o funcionamento do algoritmo, não tem problema.



Eu sugiro então que você dê uma boa reduzida na quantidade de pixels para reduzir o tempo de processamento. Se os quatro furos ali forem identificáveis na imagem resultante, já está ótimo, e poderemos passar para a próxima etapa.

Qualquer outra dúvida, é só me mandar. Se precisar, podemos marcar de conversar em algum dia dessa semana para olharmos o código.

Att,
Passarin

[Quoted text hidden]

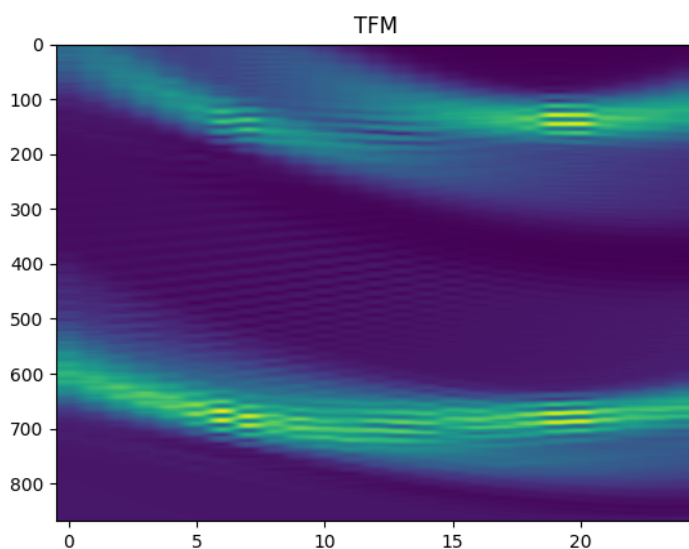
Thiago E. Kalid <thiagokkalid@gmail.com>

24 March 2021 at 11:37

To: "Thiago A. R. Passarin - UTFPR" <passarin@professores.utfpr.edu.br>

Prezado professor Passarin,

Ainda estou debugando meu código, mas até o momento o programa está consistentemente realizando o cálculo para a seguinte imagem:



Este seria o resultado conforme o recorte do problema especificado pelo senhor no e-mail anterior, apesar dos eixos não terem sido definidos para representar os valores em unidade de comprimento.

Gostaria de saber se senhor teria disponibilidade para hoje ter um breve encontro via google meet, análogo ao que tivemos semana passada?

Atenciosamente,
Thiago E. Kalid

[Quoted text hidden]

Thiago A. R. Passarin - UTFPR <passarin@professores.utfpr.edu.br>
To: "Thiago E. Kalid" <thiagokkalid@gmail.com>

24 March 2021 at 12:38

Olá, Thiago.

A imagem está condizente com o objeto inspecionado na simulação. Dá pra identificar os quatro furos. Então parece que você está no caminho certo.

Podemos conversar hoje sim. Pode ser às 14h30?

Att,
Passarin
[Quoted text hidden]

Thiago E. Kalid <thiagokkalid@gmail.com>
To: "Thiago A. R. Passarin - UTFPR" <passarin@professores.utfpr.edu.br>

24 March 2021 at 12:56

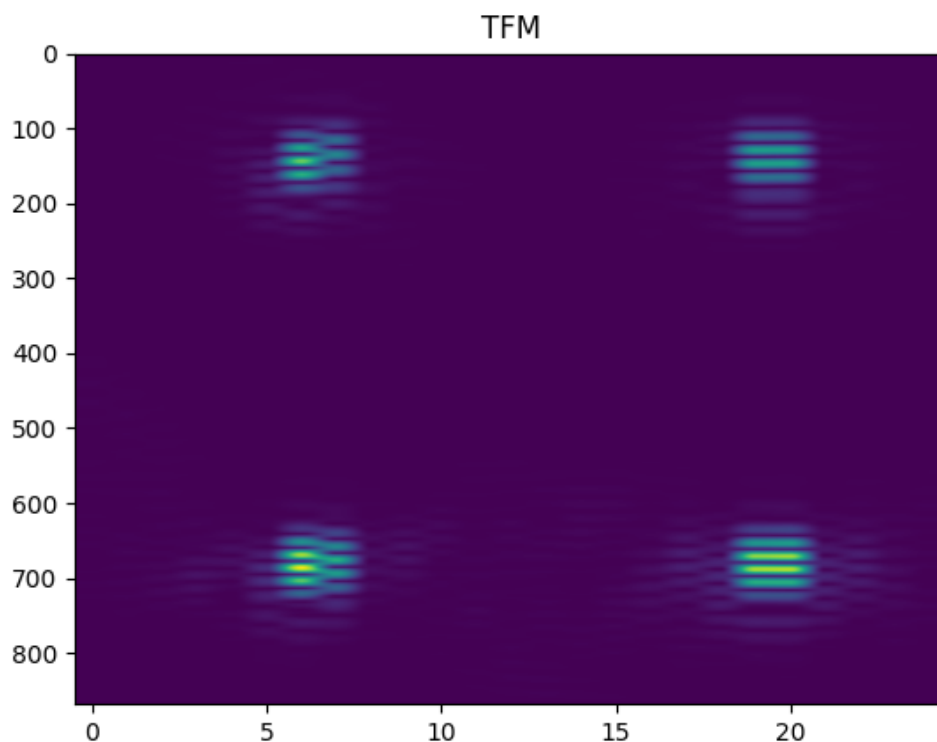
Pode ser sim professor.
[Quoted text hidden]

Thiago E. Kalid <thiagokkalid@gmail.com>
To: "Thiago A. R. Passarin - UTFPR" <passarin@professores.utfpr.edu.br>

25 March 2021 at 13:08

Prezado professor Passarin,

Depois de alguns ajustes no script cheguei a um resultado:



Gostaria de saber se para você o resultado está condizente com o esperado. Segue em anexo também o programa em si.

Atenciosamente,
Thiago E. Kalid

[Quoted text hidden]

Thiago A. R. Passarin - UTFPR <passarin@professores.utfpr.edu.br>
To: "Thiago E. Kalid" <thiagokkalid@gmail.com>

25 March 2021 at 16:39

Olá, Thiago.

O resultado está correto, é isso mesmo. Esse padrão oscilatório na representação dos refletores é causado pela frequência de oscilação (5 MHz) do pulso de ultrassom enviado. Quando conversarmos, eu te explico o método de detecção de envoltória que utilizamos para removermos esse efeito. O que acha de nos reunirmos regularmente nas segundas-feiras às 9h00? Vou sugerir para o Guilherme também, que é o outro graduando do AUSPEX que eu estou orientando diretamente.

Uma forma um pouco mais simples de remover esse efeito é fazendo a envoltória de cada coluna da imagem independentemente. Essa técnica mais simples gera alguns artefatos (ruídos) na imagem, mas nada muito grave. Se você fizer a operação `coluna = numpy.abs(scipy.signal.hilbert(coluna))` em todas as colunas, vai ver o resultado.

Att,
Passarin
[Quoted text hidden]

Thiago E. Kalid <thiagokkalid@gmail.com>
To: "Thiago A. R. Passarin - UTFPR" <passarin@professores.utfpr.edu.br>

25 March 2021 at 17:41

Prezado professor Passarin,

Para mim 9h00 das segundas-feiras está ótimo. Vou realizar a operação que você está citando para avaliar o resultado.

Atenciosamente,
Thiago E. Kalid
[Quoted text hidden]

THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>
To: MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>

7 October 2021 at 11:57

[Quoted text hidden]

MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>
To: THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>

7 October 2021 at 13:37

Boa tarde, professor.

Não estão disponíveis para mim os arquivos anexos das mensagens (nesse caso o artigo sobre o TFM e talvez um ou outro das trocas de e-mails entre vocês), só aparecem o link para o drive TFM e uma pasta zipada que acredito que seja a implementação do SAFT pelo Kalid.

Obrigado.
Mateus YM
[Quoted text hidden]

THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>
To: MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>


7 October 2021 at 14:25

Boa tarde, Mateus.

Estou enviando em anexo o artigo.

Os dados FMC e o script para carregar os dados estão na pasta do Google Drive: <https://drive.google.com/drive/folders/1W1NAJIXmbUDHEvAqKjrsWXOiwv263b6r?usp=sharing> Os nomes das variáveis criadas pelo script já estão autoexplicativos, inclusive com as unidades.

[Quoted text hidden]

 **1-s2.0-S0963869505000721-main.pdf**
581K

MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>

7 October 2021 at 14:43

To: THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>

Tranquilo, obrigado!

[Quoted text hidden]

MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>

15 October 2021 at 12:17

To: THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>

Boa tarde, professor. Tudo bem?

Estou com algumas dúvidas na resolução do TFM proposto no exercício. Antes, queria só ressaltar que eu mudei a abordagem da resolução que eu te mostrei na reunião de ontem, conversei com o Kalid pela noite e percebemos que do jeito que estava fazendo o script não considerava todos os transdutores quando eu reduzi a ROI (refiz o SAFT e ficou mais "focada" a imagem em relação a que tinha te mostrado, anexei como a figura_1 no email). Aí os índices que ele retornou para as dimensões eu só limitei dentro da range dos loops de linha/coluna e, para os pares eu considerei a faixa completa.

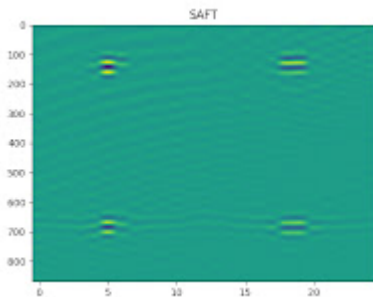
Para reduzir o tempo de debug eu escolhi somente o furo superior esquerdo para reconstruir, retornando uma matriz de tamanho (299,9) e, depois de reconstruída para esse furo, adaptar o script para a região dos 4 furos. A questão é que a figura que ele retorna ficou muito parecida com a do SAFT (seguem em anexo como o "MÉTODO UTILIZADO"_1_FURO).

Na fórmula do artigo que vocês mencionaram anteriormente, eu obtive o sinal h de maneira análoga ao meu SAFT, dentro dos loops for dos transmissores/emissores eu peguei o índice do argmin da distância relativa aos tempos e a distância relativa a soma dos Pitágoras, uma das dúvidas é que nesse caso a distância relativa aos tempo eu considerei o tempo total, ou seja fiz ($z = c \cdot t$), já que a soma dos Pitágoras engloba a distância total percorrida, aí com esse índice chamei a amplitude respectiva nos ascans como `ascans[índice,transmissor,receptor]`. E em seguida eu tenho que multiplicar esse ascan que ele retorna pela distância/velocidade? Isso eu não entendi muito bem. Já para as dimensões da ROI, nos loops for de linha/coluna eu usei para a dimensão x os respectivos valores do ensaio e para a z considerei a distância do centro do emissor até o ponto respectivo da ROI ($z = c \cdot t/2$, nesse caso). Por enquanto acredito q essas são as que ficaram em aberto.

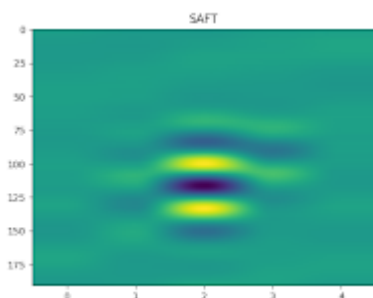
Obrigado.

[Quoted text hidden]

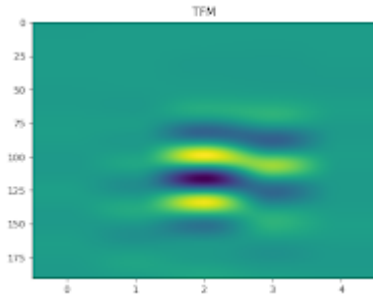
3 attachments



figura_1.tif
1201K



SAFT_1_FURO.png
49K



TFM_1_FURO.png
49K

THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>
To: MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>

15 October 2021 at 14:04

Oi Mateus. Podemos conversar agora à tarde sobre essas dúvidas? Eu estou disponível a partir das 14h40.

Deixo como sugestão o seguinte: verifique quantos sinais estão sendo somados na composição de cada pixel. No SAFT, deve ser N (número de elementos). No TFM, deve ser N ao quadrado.

[Quoted text hidden]

MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>
To: THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>

15 October 2021 at 14:08

Boa tarde, professor.

Pode ser 15:30?

Vou verificar isso que você comentou!

Obrigado.

Em 15 de out. de 2021, à(s) 14:04, THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br> escreveu:

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

Tranquilo, obrigado!

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

<Figure_9.png>

Gostaria de saber se para você o resultado está condizente com o esperado. Segue em anexo também o programa em si.

Atenciosamente,
Thiago E. Kalid

Em qua., 24 de mar. de 2021 às 12:56, Thiago E. Kalid <thiagokkalid@gmail.com> escreveu:

Pode ser sim professor.

Em qua, 24 de mar de 2021 12:38, Thiago A. R. Passarin - UTFPR
<passarin@professores.utfpr.edu.br> escreveu:

Olá, Thiago.

A imagem está condizente com o objeto inspecionado na simulação. Dá pra identificar os quatro furos. Então parece que você está no caminho certo.

Podemos conversar hoje sim. Pode ser às 14h30?

Att,
Passarin

On Wed, 24 Mar 2021 at 11:37, Thiago E. Kalid <thiagokkalid@gmail.com> wrote:
Prezado professor Passarin,

Ainda estou debugando meu código, mas até o momento o programa está consistentemente realizando o cálculo para a seguinte imagem:
<Figure_2.png>

Este seria o resultado conforme o recorte do problema especificado pelo senhor no e-mail anterior, apesar dos eixos não terem sido definidos para representar os valores em unidade de comprimento.

Gostaria de saber se senhor teria disponibilidade para hoje ter um breve encontro via google meet, análogo ao que tivemos semana passada?

Atenciosamente,
Thiago E. Kalid

Em seg., 22 de mar. de 2021 às 21:21, Thiago A. R. Passarin - UTFPR
<passarin@professores.utfpr.edu.br> escreveu:
Olá, Thiago. Vou respondendo abaixo, a partir do seu texto.

O $h_{\{tx, rx\}}$ seria obtido analogamente ao método do SAFT, isto é, o valor Y do $ascan(Y, i, j)$ que seria equivalente ao tempo necessário para a onda se propagar do transmissor i ao receptor j , para todas as combinações possíveis de transmissor e receptor. Sim, é análogo ao SAFT. Mas o tempo total corresponde ao tempo de propagação do transmissor i até o pixel em (x, z) mais o tempo de propagação do pixel em (x, z) até o receptor j . Representei graficamente essas duas parcelas do tempo em vermelho e em azul, respectivamente, no desenho abaixo.
<image.png>

Pergunto isso fiz um script que até o momento está tomando um tempo enorme para executar(ainda não consegui terminar de executá-lo), então estou confuso se a demora na execução reside na inerente complexidade computacional do programa ou em uma implementação muito ineficiente de minha parte. Sim, a complexidade computacional é muito maior, e provavelmente esse é o motivo do seu script estar levando tanto tempo. No SAFT, para um transdutor de 64 elementos, a quantidade de cálculos de distância é igual ao número de pixels vezes 64. Já no TFM, a quantidade de cálculos de distância é igual ao número de pixels vezes 64^2 , pois existem $64^2=4096$ combinações de emissor-receptor. Nós costumamos utilizar funções prontas do scipy que fazem um cálculos de distância otimizados e utilizando paralelismo (essa especificamente), e também utilizamos o numba, que tem recursos avançados de paralelismo. Para verificar o funcionamento do seu algoritmo sem precisar utilizar essas ferramentas, há duas coisas mais simples que você pode fazer:

1. Reduzir o tamanho da região reconstruída. Você pode, por exemplo, reconstruir a imagem somente em volta de alguns dos furos, como no exemplo abaixo em laranja. Se você reconstruir as coordenadas em x de -11 mm a 3 mm, e as coordenadas em z de 12 mm a 28 mm, você já pega esses quatro furos ali. Isso permite usar uma quantidade menor de pixels.

2. Reduzir a resolução espacial. Essa é outra estratégia para reduzir a quantidade de pixels e, assim, o custo computacional. Isso vai prejudicar a qualidade da imagem, mas por enquanto, como o objetivo é entender o funcionamento do algoritmo, não tem problema.

<image.png>

Eu sugiro então que você dê uma boa reduzida na quantidade de pixels para reduzir o tempo de processamento. Se os quatro furos ali forem identificáveis na imagem resultante, já está ótimo, e poderemos passar para a próxima etapa.

Qualquer outra dúvida, é só me mandar. Se precisar, podemos marcar de conversar em algum dia dessa semana para olharmos o código.

Att,
Passarin

On Mon, 22 Mar 2021 at 18:40, Thiago E. Kalid <thiagokkalid@gmail.com> wrote:

Prezado professor Thiago Passarin,

Gostaria de esclarecer se eu estou compreendendo ou não o método TFM.

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

<Figure_1.png>

<Figure_2.png>

[Quoted text hidden]

THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>
To: MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>

15 October 2021 at 14:43

Pode ser. Deixamos marcado então.

[Quoted text hidden]

MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>
To: THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>

15 October 2021 at 14:56

Ok, no mesmo link da reunião de ontem?

[Quoted text hidden]

THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>
To: MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>

15 October 2021 at 15:16

Sim, no mesmo de ontem: <https://meet.google.com/new-qgkc-ziw> Coloquei na agenda também.

[Quoted text hidden]

MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>
To: THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>

22 October 2021 at 13:08

Boa tarde, professor. Tudo bem?

Segue em anexo o resultado do script para o TFM com os laços for, como comentamos ontem na reunião eu plotei o resultado para o TFM em módulo que estava faltando. Aproveitei e coloquei os resultados do script do SAFT também

para o ensaio FMC, mostrando meio que uma "comparação" dos dois. Fiquei com dúvida na parte que vocês conversaram anteriormente sobre "coluna = numpy.abs(scipy.signal.hilbert(coluna)) ". Acredito que a resolução inicial do TFM seja essa, agora tenho que ver toda questão dos arrays e cdist que conversamos.

Você quer que eu te envie o script também?

Obrigado e bom final de semana!

Mateus

[Quoted text hidden]



SAFT_TFM_FMC_RESOLUÇÃO.pdf

187K

THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>
To: MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>

23 October 2021 at 15:58

Olá, Mateus.

Legal, os resultados parecem corretos, com o TFM gerando menos artefatos do que o SAFT. Não é necessário me enviar o script;

Com relação à operação `coluna = numpy.abs(scipy.signal.hilbert(coluna))`, a ideia é tirar essa característica oscilatória da imagem e deixá-la mais suave. Essa operação deve ser feita coluna a coluna na imagem, gerando uma nova imagem onde as colunas correspondem às envoltórias das colunas originais, sem a oscilação. Essa operação é chamada de demodulação, ou detecção de envelope. Fiz um pequeno exemplo de demodulação de um sinal arbitrário para demonstrar a ideia: <https://colab.research.google.com/drive/1ouewRk2yhCtD1wFCN4ZbFk74csJVzvyf?usp=sharing> Eu sugiro fazer o plot assim (unidimensional) de algumas colunas da imagem para visualizar bem a operação.

Com relação ao uso do `cdist`, aquele primeiro exercício que eu sugeri na reunião com certeza vai ajudar a compreender como se utiliza essa função. O exercício é: dada uma ROI, gerar uma imagem que mostre (não o resultado da reconstrução, mas) as distâncias entre cada pixel e algum ponto fixo do espaço. Esse ponto fixo pode ser, por exemplo, o primeiro elemento do transdutor. Sugiro chamar a função `plt.colorbar()` após a criação da imagem, pois ela exibe uma barra com a escala de cores.

Bom fim de semana!

[Quoted text hidden]

MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>
To: THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>

25 October 2021 at 16:39

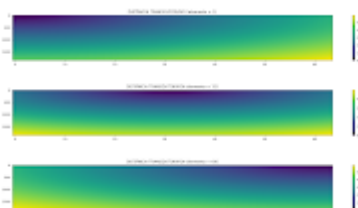
Boa tarde, professor. Tudo bem?

Obrigado pelo exemplo do script vou aplicar no TFM e qualquer coisa conversamos. Sobre o `cdist`, fiz uma implementação inicial do exercício proposto com base nos dados do ensaio do TFM mesmo, para o primeiro transdutor, o do meio e o último na faixa dos 64 elementos. Pelo que eu entendi na reunião era a transição das cores que devia ser observada porém, a disposição delas ficou trocada pelo que conversamos. Anexei a figura no email para você dar uma olhada.

Obrigado,

Mateus

[Quoted text hidden]



dist_elem_roi.png

180K

THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>
To: MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>

25 October 2021 at 17:51

Olá, Mateus.

Está correta a disposição das cores nas imagens. Eu é que tinha me enganado: mas próximo do elemento de referência, o valor é menor, o que corresponde a um tom mais azul.

Para gerar uma dessas imagens, você chama o `cdist` várias vezes dentro de laços `for`, ou monta as matrizes e chama o `cdist` uma única vez?

Att,
Passarin
[Quoted text hidden]

MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>
To: THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>

25 October 2021 at 18:32

Tranquilo, sem problemas!

Para gerar a imagem eu montei as matrizes e chamei o `cdist` uma vez, ela retornou uma matriz (64,118912) e usei as funções que você recomendou `np.reshape` e `np.transpose` para retornar (1858,64) para cada um dos 64 elementos.

Obrigado,
Mateus
[Quoted text hidden]

MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>
To: THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>

25 October 2021 at 18:35

Tranquilo, sem problemas!

Para gerar a imagem eu montei as matrizes e chamei o `cdist` uma vez, ela retornou uma matriz (64,118912) e usei as funções que você recomendou `np.reshape` e `np.transpose` para retornar (1858,64) para cada um dos 64 elementos. Usei como referência também umas partes da função `ImagingROI` do framework.

Obrigado,
Mateus
[Quoted text hidden]

THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>
To: MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>

27 October 2021 at 13:00

A ideia agora é você utilizar essa matriz para pegar as distâncias entre cada elemento e cada pixel, dentro dos laços `for` do algoritmo. Isso vai substituir o cálculo de distância. Como todas as distâncias foram calculadas em "batelada", com uma função mais eficiente, o tempo total de execução deve se reduzir consideravelmente.

[Quoted text hidden]

MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>
To: THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>

29 October 2021 at 15:20

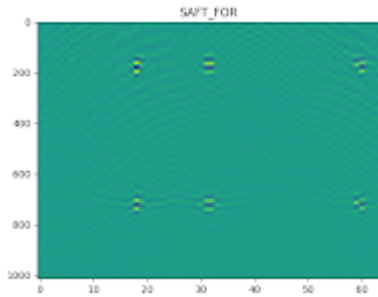
Boa tarde, professor. Tudo bem?

Com as alterações que conversamos ontem, refiz o SAFT para uma ROI maior do ensaio FMC e comparei o tempo em que as duas levam para gerar a mesma reconstrução (1013,64). A diferença é absurda mesmo, com o `cdist` e sem aquele caminhaço de prints que estava usando para acompanhar o código, ele roda mais ou menos 2~3 vezes mais rápido. Calculando a distância dentro dos loops e sem os prints obtive algo em torno de 01:05,0 já para o CDIST algo próximo de 00:24,0.

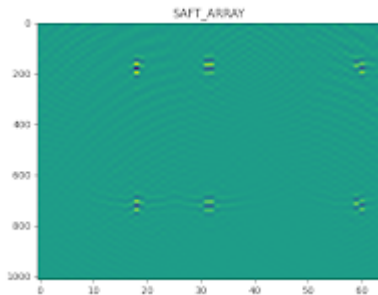
Vou ver o TFM agora e vou te atualizando.

Obrigado e bom final de semana/feriado.
[Quoted text hidden]

2 attachments



SAFT_ARRAY.png
106K



SAFT_FOR.png
106K

MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>
To: THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>

29 October 2021 at 18:57

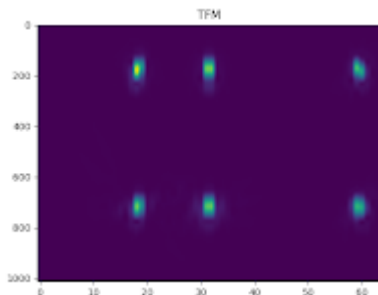
Boa noite, professor.

Anexei neste email o resultado do TFM com o CDIST e a função da envoltória (hilbert) que agora acredito que esteja sendo aplicada corretamente. Complementando o que eu falei no email anterior, novamente o algoritmo foi MUITO mais rápido mesmo, pra uma ROI bem maior (1013,64) o algoritmo demorou algo em torno dos 25 minutos, enquanto a implementação original (869,25) calculando a distância dentro dos laços demorou demais, não lembro exatamente mas foram horas rodando.

Obrigado.

Mateus

[Quoted text hidden]



TFM.png
33K

MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>
To: THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>

8 November 2021 at 15:21

Boa tarde, professor. Tudo bem?

Montei a matriz com os valores complexos, como conversamos na quinta passada, chamando o hilbert nos ascans e depois fiz o TFM com esse sinal, para no final calcular o módulo da matriz da imagem.

[Quoted text hidden]

MATEUS YAMADA MULLER <mateusyamada@alunos.utfpr.edu.br>
To: THIAGO ALBERTO RIGO PASSARIN <passarin@professores.utfpr.edu.br>

8 November 2021 at 15:44

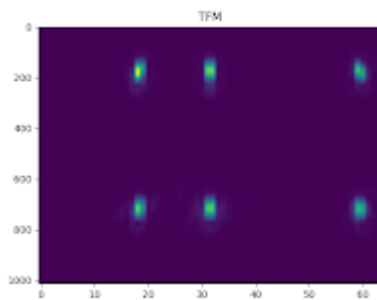
Boa tarde, professor. Tudo bem?

Montei a matriz com os valores complexos, como conversamos na quinta passada, chamando o hilbert nos ascans e

depois fiz o TFM com esse sinal, para no final calcular o módulo da matriz da imagem. Acontece que a imagem que o script retornou é praticamente a mesma da calculando a envoltória da imagem final, anexei ela no email. Não tenho muita certeza de onde possa estar o erro dessa vez, você teria alguma ideia?

Obrigado,
Mateus

[Quoted text hidden]



teste_hilbert.png
33K