

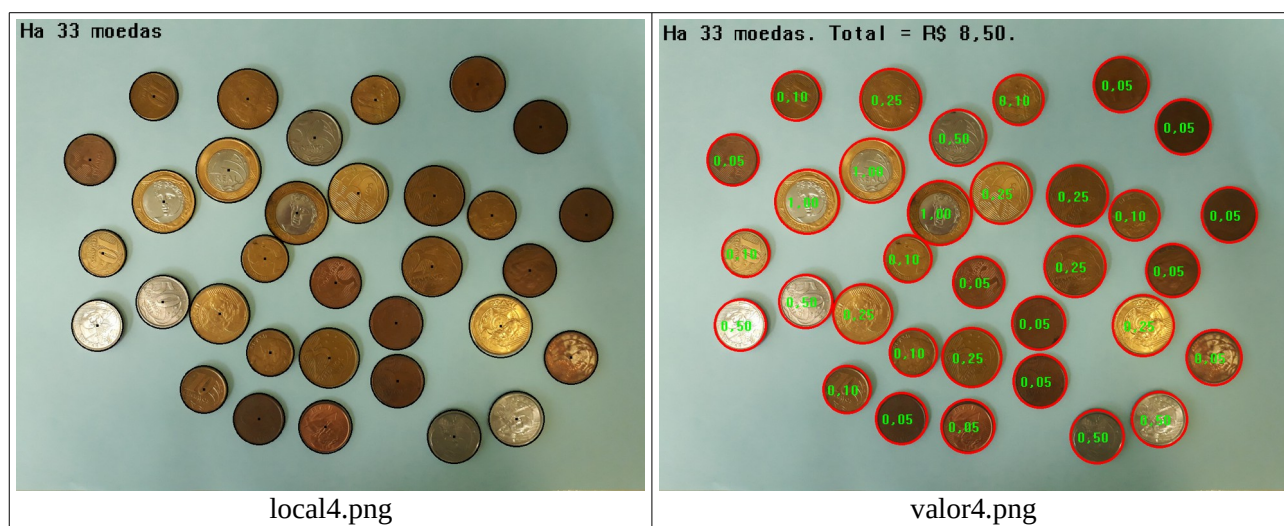
**Obs. 1:** Cada dia de atraso acarreta perda de 1 ponto no exercício.

**Obs. 2:** Este EP deve ser resolvido individualmente. EPs iguais receberão nota zero.

O objetivo deste exercício é fazer um programa C/C++ ou Python que, dada uma imagem de moedas brasileiras atuais sobre fundo de papel azul, conta quantas moedas tem na imagem e localiza espacialmente cada moeda (veja a imagem local4.png abaixo). Fazendo isto, a nota máxima deste EP será 10. Opcionalmente, o seu programa pode identificar o valor de cada moeda e calcular o valor total das moedas na imagem (veja a imagem valor4.png abaixo). Fazendo isto, a nota máxima deste EP será 12.

As moedas podem estar "grudadas" entre si, mas não sobrepostas uma em cima da outra.

Procurem resolver este exercício **sem usar transformada de Hough**, pois provavelmente não veremos esta técnica nesta disciplina. O exercício foi pensado para ser resolvido usando filtro linear e/ou casamento de modelos (template matching).



A forma de chamar o programa deve ser obrigatoriamente:

```
linux$ ep1 moeda1.jpg saida1.png
```

```
windows> ep1 moeda1.jpg saida1.png
```

O programa deve emitir um aviso amigável se o usuário entrar argumentos inválidos (número de parâmetros incorreto, arquivo inexistente, etc).

As quatro imagens de entrada que você deve testar estão em:

<http://www.lps.usp.br/hae/psi5796/ep1-2020/moeda.zip>

Estas imagens foram tiradas por um celular comum, no modo "pro" (com todos os ajustes manuais). Especialmente, desliguei WB (white balance) automático. Ligando WB, o fundo azul torna-se branco.

As imagens de saída (para servirem de exemplo) estão em:

<http://www.lps.usp.br/hae/psi5796/ep1-2020/localizacao.zip>

<http://www.lps.usp.br/hae/psi5796/ep1-2020/valor.zip>

A figura ao lado mostra todas as moedas de reais: na parte superior a antiga família de moedas (produzidas entre 1994-1998) e na parte inferior a nova família (produzidas a partir de 1998). Vamos supor que na imagem só aparecem moedas da nova família. Também vamos supor que não aparece moeda de 1 centavo, pois esta parou de ser produzida desde 2005. Assim, na imagem só podem aparecer moedas de 5, 10, 25, 50 centavos e de 1 real, da nova família. Para possibilitar identificar os valores das moedas pelos tamanhos, vamos supor que em cada imagem aparece pelo menos uma moeda de 1 real.



Como localizei as moedas (você pode resolver de outro jeito):

1) Diminuí a resolução da imagem de entrada e apliquei algum filtro de suavização. Depois, mapeei os pixels com cor de fundo para branco e os pixels com cor de moeda para preto.

2) Criei um modelo de círculo, preto por dentro e branco por fora. Redimensionei o modelo para diferentes escalas, usando progressão geométrica.

3) Procurei todos os modelos redimensionados na imagem, usando casamento de modelos. Com isso, cada pixel possui vários valores de correlação (um valor para cada modelo). Para cada pixel, procurei o maior valor de correlação. Se esse valor for maior que um certo limiar e for maior que os valores dos vizinhos, o pixel é candidato a ser o centro de uma moeda.

4) Eliminei as falsas detecções eliminando os pixels-candidatos que estão muito próximos de outros pixels-candidatos.

Os trabalhos abaixo descrevem melhor a ideia acima (denominada de "granulometria por correlação"), mas você não precisa lê-los:

[\[EAAI2015\]](#) S. A. de Araújo, J. H. Pessota and H. Y. Kim, "Beans Quality Inspection Using Correlation-Based Granulometry," *Engineering Applications of Artificial Intelligence*, vol. 40, pp. 84-94, April 2015.

[\[JPM2013\]](#) H. Y. Kim, R. H. Maruta, D. R. Huanca and W. J. Salcedo, "Correlation-Based Multi-Shape Granulometry with Application in Porous Silicon Nanomaterial Characterization," *Journal of Porous Materials*, April 2013, Vol. 20, No. 2, pp 375-385.

[\[ECST2010a\]](#) R. H. Maruta, H. Y. Kim, D. R. Huanca and W. J. Salcedo, "A New Correlation-Based Granulometry Algorithm with Application in Characterizing Porous Silicon Nanomaterials," *ECS Transactions*, vol. 31, pp. 273-280, 2010.

Como calculei os valores das moedas (você pode resolver de outro jeito):

1) Procurei as moedas de 1 real - que são fáceis de reconhecer pois possuem duas cores e são as maiores moedas da imagem. Estou supondo que cada imagem possui pelo menos uma moeda de 1 real. Calculei o raio médio das moedas de 1 real na imagem dada.

2) Identifiquei as moedas de 50 centavos pela cor.

3) Identifiquei as moedas de 25, 10 e 5 centavos pelo raio (em relação ao raio de moeda de 1 real).

**Obs. 1:** Pode usar (se quiser) a biblioteca Cekeikon/OpenCV.

**Obs. 2:** Entregue o programa-fonte (ep1.cpp ou ep1.py) e um documento PDF (relatorio.pdf) com no máximo 4 páginas com os comentários descrevendo o funcionamento do programa. O envio do relatório é obrigatório (veja o anexo).

(a) Se você fez o programa no ambiente usado na classe (Cekeikon/OpenCV, Windows/Linux), basta entregar o programa-fonte ep1.cpp ou ep1.py para poder corrigir o seu programa.

(b) Se você quiser usar alguma biblioteca diferente ou programar em outro ambiente, converse antes com o professor. Neste caso, você deve combinar com o professor um horário para mostrar o seu programa funcionando.

**Obs. 3:** Compacte todos os arquivos como nome\_sobrenome.zip e envie o arquivo via email para hae@lps.usp.br. Dentro do prazo, você pode substituir o arquivo anterior por um novo. Só o último arquivo entregue será corrigido.

### **Anexo: Relatórios dos exercícios programas**

O mais importante numa comunicação escrita é que o leitor entenda, sem esforço e inequivocamente, o que o escritor quis dizer. O texto ficar "bonito" é um aspecto secundário. Se uma (pseudo) regra de escrita dificultar o entendimento do leitor, essa regra está indo contra a finalidade primária da comunicação. No site do governo americano [1], há regras denominadas de "plain language" para que comunicações governamentais sejam escritas de forma clara. As ideias por trás dessas regras podem ser usadas em outros domínios, como na escrita científica. Resumo abaixo algumas dessas ideias.

(1) Escreva para a sua audiência. No caso do relatório, a sua audiência será o professor ou o monitor que irá corrigir o seu exercício. Você deve focar na informação que o seu leitor quer conhecer. Não precisa escrever informações que são inúteis ou óbvias para o seu leitor.

(2) Organize a informação. Você é livre para organizar o relatório como achar melhor, porém sempre procurando facilitar o entendimento do leitor. Seja breve. Quebre o texto em seções com títulos claros. Use sentenças curtas. Elimine as frases e palavras que podem ser retiradas sem prejudicar o entendimento. Use sentenças em ordem direta (sujeito-verbo-predicado).

(3) Use palavras simples. Use o tempo verbal o mais simples possível. Evite cadeia longa de nomes, substituindo-os por verbos (em vez de "desenvolvimento de procedimento de proteção de segurança de trabalhadores de minas subterrâneas" escreva "desenvolvendo procedimentos para proteger a segurança dos trabalhadores em minas subterrâneas"). Minimizar o uso de abreviações (para que o leitor não tenha que decorá-las). Use sempre o mesmo termo para se referir à mesma realidade (pode confundir o leitor se usar termos diferentes para se referir a uma mesma coisa). O relatório não é obra literária, não tem problema repetir várias vezes a mesma palavra.

(4) Use voz ativa. Deixe claro quem fez o quê. Se você utilizar oração com sujeito indeterminado ou na voz passiva, o leitor pode não entender quem foi o responsável (Ex: "Criou-se um novo algoritmo" - Quem criou? Você? Ou algum autor da literatura científica?). O site diz: "Passive voice obscures who is responsible for what and is one of the biggest problems with government writing."

(5) Use exemplos, diagramas, tabelas, figuras e listas. Ajudam bastante o entendimento.

---

1 <https://plainlanguage.gov/guidelines/>

## **O relatório deve conter pelo menos as seguintes informações:**

### *Identificação*

Nome, número USP, nome da disciplina, etc.

### *Breve enunciado do problema*

Apesar do enunciado do problema ser conhecido ao professor/monitor, descreva brevemente o problema que está resolvendo. Isto tornará o documento compreensível para alguma pessoa que não tem o enunciado do EP à mão.

### *Técnica(s) utilizada(s) para resolver o problema*

Descreva quais técnicas você usou para resolver o problema. Se você mesmo inventou a técnica, descreva a sua ideia, deixando claro que a ideia foi sua. Se você utilizou alguma técnica já conhecida, utilize o nome próprio da técnica (por exemplo, filtragem Gaussiana, algoritmo SIFT, etc.) juntamente com alguma referência bibliográfica onde a técnica está descrita. Use elementos gráficos como imagens intermediárias e diagramas, pois ajudam muito a compreensão. Não "copie-e-cole" código-fonte, a não ser que seja relevante. Use preferencialmente o pseudo-código.

### *Ambiente de desenvolvimento utilizado*

Em qual plataforma você desenvolveu o programa? Como o professor/monitor pode compilar o programa? Você utilizou que bibliotecas?

### *Operação*

Como o professor/monitor pode executar o programa? Que argumentos são necessários para a execução do programa? Há parâmetros que devem ser configurados? Quais arquivos de entrada são necessários? Quais arquivos de saída são gerados?

### *Resultados Obtidos*

Descreva os resultados obtidos. Qual é o tempo de processamento típico? O problema foi resolvido de forma satisfatória?

### *Referências*

Descreva o material externo utilizado, como livros/artigos consultados, websites visitados, etc.