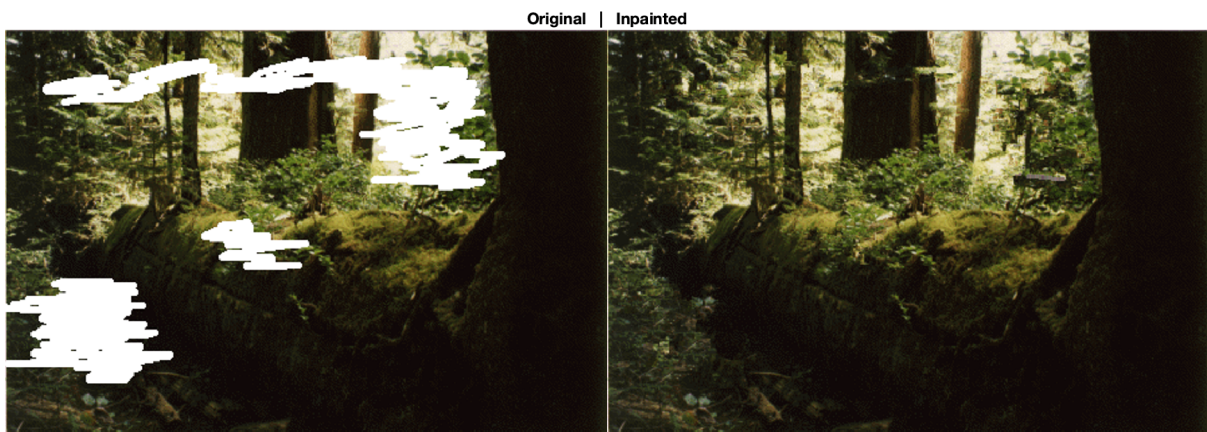


Vkresľovanie do obrazu

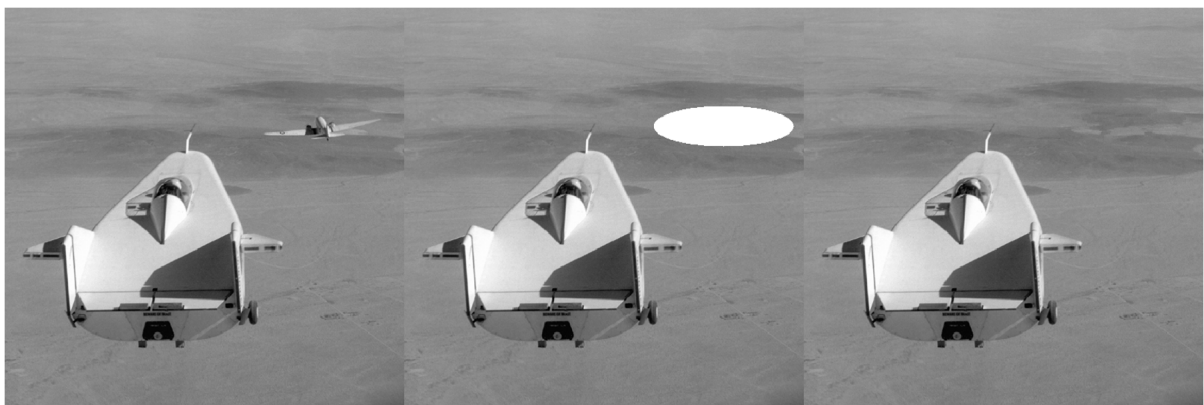
Je viacero prípadov keď je vhodné do obrazu vkresľovať (inpainting). Môže sa jednať o odstraňovanie nežiadúcich objektov z obrazu či cieleľnú úpravu obsahu. Vtedy hovoríme o **retušovaní**. Môže sa však jednať aj o opravu poškodených či doplnenie zničených či chýbajúcich častí obrazu, t.j. jedná sa o **rekonštrukciu** (restoration) obrazu. Dôležité je, že zmena nastane **vizuálne hodnoverným** spôsobom. Pri vyplňaní chýbajúcich častí sa z matematického hľadiska sa jedná o interpolačný problém.

V tejto časti si ukážeme si dve metódy vkresľovania: vkresľovanie na základe príkladov (`inpaintExemplar`) a koherentný prenos (`inpaintCoherent`).

Príklady použitia oboch metód v prostredí MATLAB sú na Obr. 1 - Obr. 4, zdrojové kódy použité na ich vygenerovanie sú uvedené v prílohe.



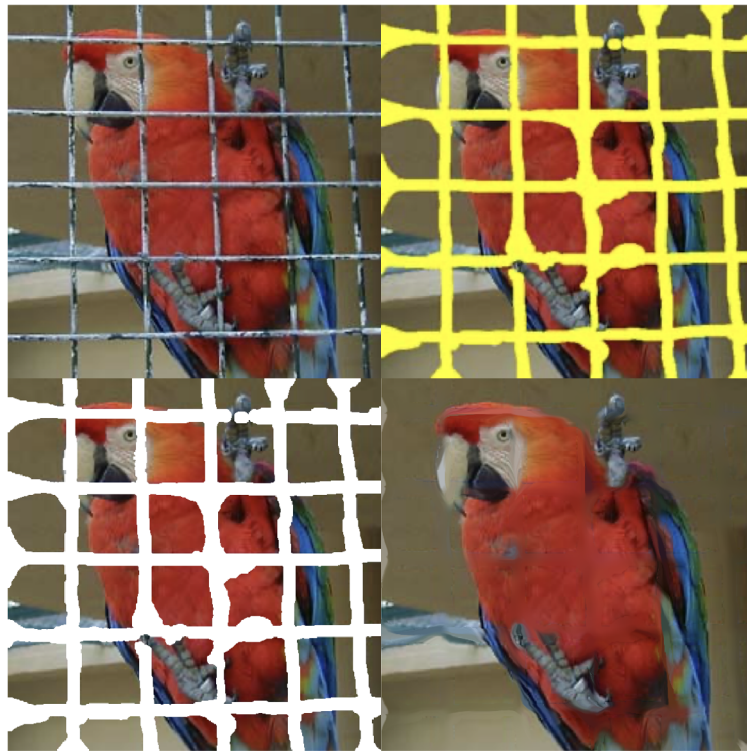
Obr. 1 Príklad vkresľovania pomocou exemplar based metódy. Vľavo poškodený obraz (sú prítomné biele plochy, ktoré je treba dokresliť), vpravo rekonštruovaný obraz.



Obr. 2 Príklad vkresľovania pomocou exemplar based metódy. Vľavo originálny obraz, v strede je biela oblasť namiesto objektu, ktorý chceme odstrániť, vpravo výsledný vyretušovaný obraz



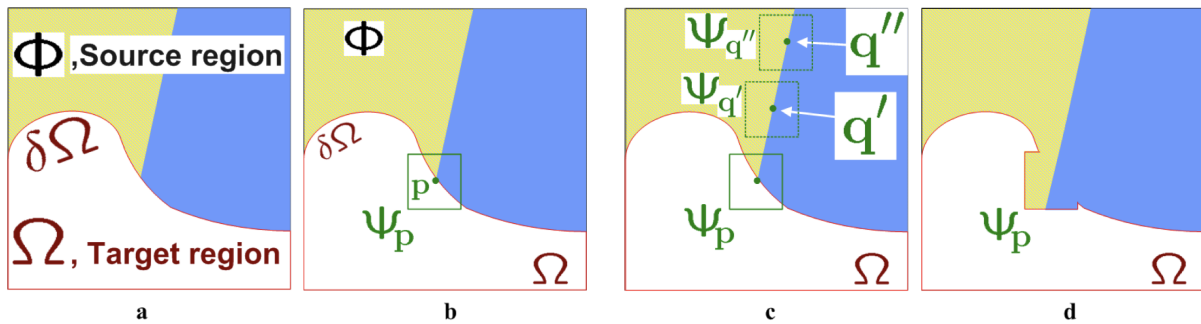
Obr. 3 Príklad vkresľovania pomocou koherentného prenosu. Vľavo originálny obraz, v strede je biela oblasť namiesto objektu, ktorý chceme odstrániť, vpravo výsledný vyretušovaný obraz



Obr. 4 Príklad vkresľovania pomocou koherentného prenosu. Cieľom je zbaviť sa mreží. Vľavo hore originálny obraz, vpravo hore manuálne vyznačená oblasť mreže žltou farbou, dole vľavo prahovaním a dilatáciou vymazaná označená časť, vpravo dole vkreslené chýbajúce časti.

Pri metóde koherentného prenosu [1] sa pracuje na úrovni bodov obrazu. Pri vkresľovaní do oblasti sa začína sa na hranici oblasti a pokračuje sa smerom do jej vnútra. Vkresľovaná hodnota bodu sa odhaduje z okolia daného bodu ako váhovaný priemer, pričom sa zvažuje buď smerový alebo difúzny prenos informácie z okolia.

Pri metóde vkresľovania na základe príkladov [2] sa pracuje so záplatami (patches) čo sú malé obdĺžnikové oblasti. V rámci algoritmu sa postupne identifikujú najvhodnejšie miesta na vloženie záplaty, pričom stred záplaty leží na kraji cieľovej oblasti. Vhodná záplata sa hľadá v okolí cieľovej oblasti na základe podobnosti (metrika sum of square difference - SSD). Cieľová oblasť sa takto postupne znižuje. Princíp vkladania záplaty je naznačený na Obr. 4.



Obr. 4 Princíp vkladania záplaty na okraj cieľovej oblasti pri metóde vkreslovania na základe príkladov. Jednotlivé obrázky a, b, c, d označujú fázy algoritmu pri vkladaní. Fáza a označuje východiskový situáciu. Vo fáze b, je identifikované najvhodnejšie cieľové miesto p na vloženie záplaty Ψ_p (zelený štvorček). Následne vo fáze c sú nájdené vhodné zdroje pre záplatu. Najpravdepodobnejšie ležia niekde na hrane, ktorá vedie do oblasti Ψ_p . Sú to napr. záplaty $\Psi_{q'}$, alebo $\Psi_{q''}$. Následne vo fáze d je vložená najvhodnejšia záplata na cieľové miesto.

Referencie

- [1] F. Bornemann and T. März. "Fast Image Inpainting Based on Coherence Transport." Journal of Mathematical Imaging and Vision. Vol. 28, 2007, pp. 259–278.
- [2] Criminisi, A., P. Perez, and K. Toyama. "Region Filling and Object Removal by Exemplar-Based Image Inpainting." IEEE Transactions on Image Processing. Vol. 13, No. 9, 2004, pp. 1200–1212.

Prílohy

Zdrojový kód programu, ktorý vytvoril Obr. 1

```
clear all; close all; clc;
I = imread('INPUT/forestdistorted.png');
mask = imread('INPUT/imagemask.png');

I(repmat(mask, [1, 1, 3]) > 0) = 255;
J = inpaintExemplar(I, mask, 'PatchSize', 7);
montage({I, J});
title(['Original', ' | ', 'Inpainted'])
exportgraphics(gcf(), 'inpaintExemplarExample.png')
```

Zdrojový kód programu, ktorý vytvoril Obr. 2

```
clear all; close all; clc;
I = imread('INPUT/forestdistorted.png');
mask = imread('INPUT/imagemask.png');

I(repmat(mask, [1, 1, 3]) > 0) = 255;
J = inpaintExemplar(I, mask, 'PatchSize', 7);
montage({I, J});
title(['Original', ' | ', 'Inpainted'])
exportgraphics(gcf(), 'inpaintExemplarExample.png')
```

Zdrojový kód programu, ktorý vytvoril Obr.3

```
clear all; close all; clc;
I = imread('cameraman.tif');
imshow(I, []);
```

```

%h = drawellipse('Center',[233 160],'SemiAxes',[18 25]);
h = drawrectangle('Position',[190 102 16 52]);
mask = createMask(h);
IM=I;
IM(mask>0)=255;
J = inpaintCoherent(IM,mask,"Radius",1);
montage({I,IM,J},"Size",[1,3]);
exportgraphics(gcf(),['inpaintCoherentExample' ...
'.png'])

```

Zdrojový kód programu, ktorý vytvoril Obr. 4

```

clear all; close all; clc;
I = imread('INPUT/parrot550.png');
M = imread('INPUT/parrotMasked550.png');
%identify yellow points
YP = M(:, :, 1) ≥ 200 & M(:, :, 2) > 200 & M(:, :, 3) ≤ 200;
SE = strel("square",4);
%add some safe margin - dilate the mask
YPD = imdilate(YP,SE);
Basis=I;
Basis( repmat(YPD,[1,1,3]) )=255;
J = inpaintCoherent(I,YPD,"Radius",14,"SmoothingFactor",7);
montage({I,M,Basis,J},"Size",[2,2]);
exportgraphics(gcf(), 'inpaintCoherentParrot.png')

```