

## Lösungsidee

In dieser Aufgabe muss man nur überprüfen, ob nebeneinanderliegende Pixel die gleiche Farbe haben. Wenn das der Fall ist, werden sie weiß gefärbt.

Mithilfe von Zählwiederholungen kann jeder Pixel eines Bildes überprüft werden und falls nötig mit allen gleichfarbigen Nachbarn gefärbt werden.

Dadurch werden natürlich auch unnötige Pixel weiß gefärbt. Allerdings steht in der Aufgabenstellung „Pixel..., die zu einem Rhinozelfanten gehören könnten“. Somit sollte man auch andere Pixel markieren dürfen.

Dennoch gibt es eine Möglichkeit, die Zahl der fälschlich gefärbten Pixel zu reduzieren: Wenn man annimmt, dass eine Schuppe sowohl in horizontale, als auch in vertikale Richtung größer als ein Pixel ist, kann man nur solche Pixel weiß färben, die sowohl bei der horizontalen, als auch bei der vertikalen Untersuchung weiß gefärbt worden sind. Dadurch sieht man die Rhinozelfanten aber nicht mehr so deutlich. Dennoch sind sie bei beiden Möglichkeiten klar erkennbar.

## Beispiele

Zunächst die Beispiele mit dem ersten Verfahren:



Und nun die Bilder mit der zweiten Möglichkeit:



## Quelltext

Horizontale Überprüfung des Bildes nach gleichfarbigen Nachbarn, vertikal funktioniert ähnlich

```
private BufferedImage imgHorizontal(BufferedImage img)
```

```
{
    //Untersuchung aller Pixel
    for(int i=0; i<img.getHeight(); i++)
    {
        for(int j=0; j<img.getWidth()-1; j++)
        {
            //RGB-Wert von weiß
            int rgb = new Color(255,255,255).getRGB();
            //Anzahl gleichfarbiger Pixel in einer Kette
```

```

    int k=0;
    //Wenn der folgende Pixel die gleiche Farbe hat, k inkrementieren und den nächsten Pixel anschauen
    while(j+k+1<img.getWidth() && img.getRGB(j+k,i) == img.getRGB(j+k+1,i))
    {
        k++;
    }
    //Wenn es gleiche Pixel gibt, den ersten und anschließend alle weiteren Pixel weiß färben
    if(k>0)
    img.setRGB(j,i,rgb);
    for(int l=0; l<k; l++)
    {
        img.setRGB(j+l,i,rgb);
    }
    //Den nächsten nicht gefärbten Pixel anschauen
    j+=k;
}
}
return img;
}

```

Überprüfung, ob die zu speichernden Bilder an der Position (x,y) weiß gefärbt werden sollen

//Wenn in beiden Bearbeitungen Pixel weiß ist, dann werden beide Originalbilder bearbeitet

```

if(imgH.getRGB(j,i) == rgb && imgV.getRGB(j,i) == rgb)
{
    imgAnd.setRGB(j,i,rgb);
    imgOr.setRGB(j,i,rgb);
}
//Wenn nur bei einem, wird nur eins weiß gefärbt
else if(imgH.getRGB(j,i) == rgb || imgV.getRGB(j,i) == rgb)
{
    imgOr.setRGB(j,i,rgb);
}
}

```