

## 4 Técnicas Desenvolvidas

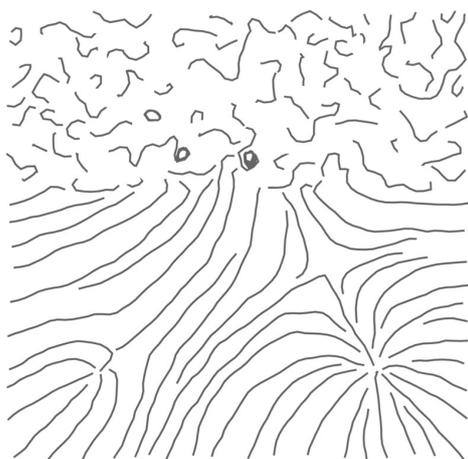
Neste capítulo descrevemos as técnicas desenvolvidas neste trabalho. Apresentamos uma interface para a *edição* de campos vetoriais (18) e como construir a *segmentação* deste baseado em sua topologia. Além disso, mostraremos como essas etapas auxiliam na visualização por imagens auto-animadas.

### 4.1 Edição de Campos Vetoriais - Remoção de Ruidos

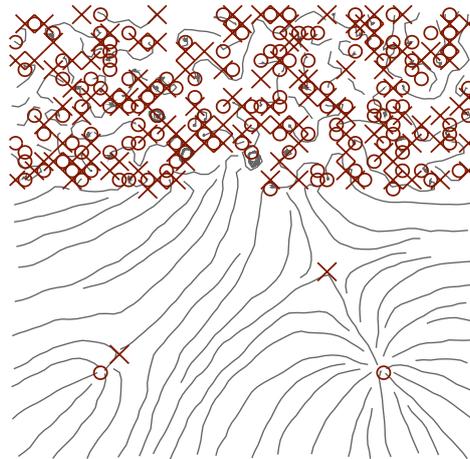
Vários fatores podem interferir na visualização dos campos vetoriais. Dados reais estão sujeitos a variações causadas por ruídos devido principalmente às imprecisões provenientes das medições.

Apresentamos uma metodologia que consiste em permitir ao usuário selecionar localmente uma escala de ruído para remoção, definindo um parâmetro de escala  $s(x, y)$  em cada ponto (18).

A Figura 4.1 ilustra a técnica que estamos propondo. Construímos um campo vetorial artificial e adicionamos ruído não gaussiano em escalas diferentes, sendo em maior escala na parte superior e menor escala na parte inferior. As singularidades da parte inferior do campo devem ser mantidas, já as da parte superior, com exceção de um poço, devem ser removidas.



4.1(a): Campo vetorial artificial



4.1(b): Detecção de singularidades

Figura 4.1: Campo vetorial artificial representado por linhas de fluxo.

A primeira etapa, consiste na geração do espaço de escala a partir do campo original, onde o usuário escolhe uma escala central  $s_0$  (ver Figura 4.2). Para evitar o custo de definir o parâmetro de escala  $s(x, y)$  em todos os pontos amostrados no campo, apresentamos para o usuário as singularidades que aparecem e/ou desaparecem em escalas vizinhas de  $s_0$  de acordo com um parâmetro definido pelo usuário. Este parâmetro pode ser a diferença entre escalas ou a quantidade de mudanças topológicas. Neste trabalho realizamos a filtragem por *random walk* com núcleo Gaussiano  $G_\alpha$  e com o núcleo anisotrópico  $A_{\sigma,\tau}$  (3, 19):

$$A_{\sigma,\tau}(x, y, \mathbf{v}) = \exp\left(-\frac{x^2 + y^2}{2\sigma}\right) \exp\left(-\frac{\|\mathbf{v}\|^2}{2\tau}\right)$$

Ambos núcleos representam naturalmente o dado na forma hierárquica do espaço de escala.

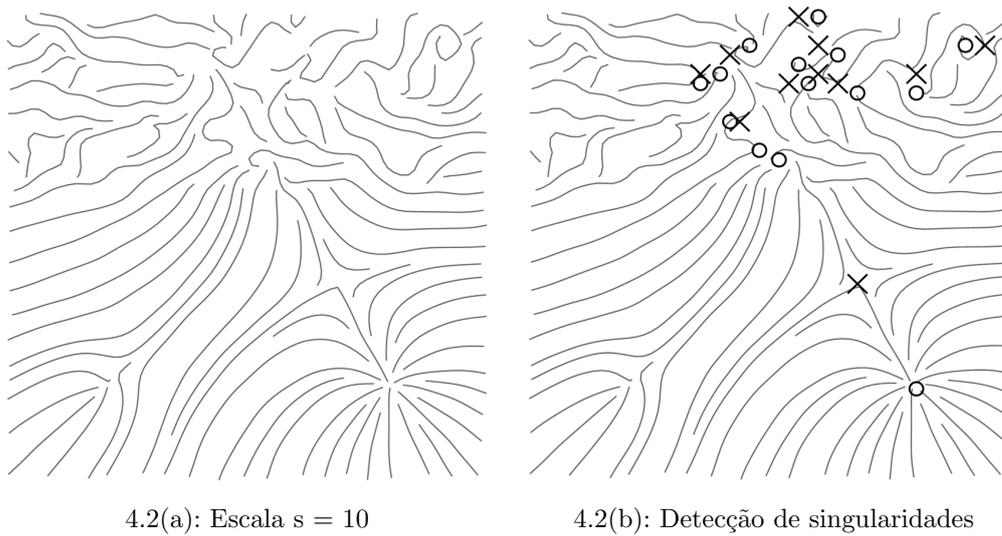
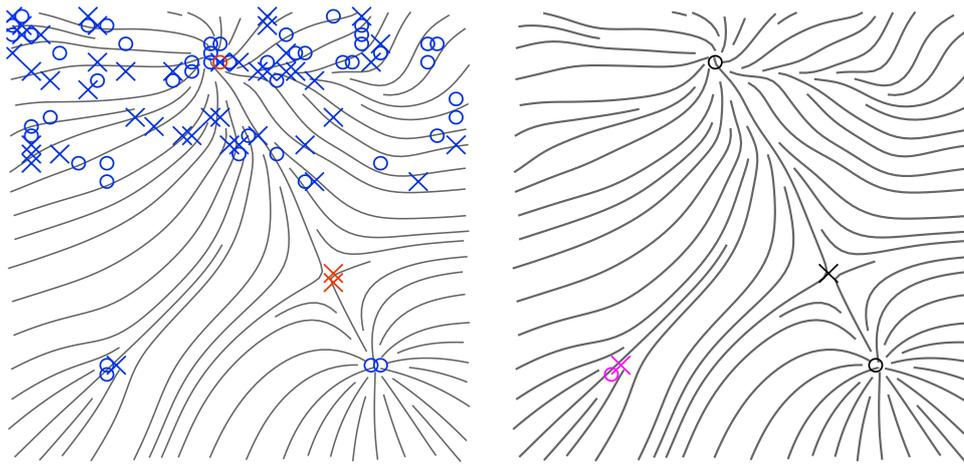


Figura 4.2: Campo vetorial artificial na escala  $s = 10$  do espaço de escala Gaussiano.

Em posse do espaço de escala, o usuário pode decidir quais mudanças topológicas são relevantes e desejadas para o campo em questão (ver Figura 4.3). Quando uma mudança topológica em um ponto singular  $(x_0, y_0)$  é selecionada, definimos  $s(x_0, y_0)$  para ser a escala mais próxima de  $s_0$  que reverte a mudança. Em outras palavras, utilizamos o espaço de escala para permitir que o usuário decida que escala utilizar *localmente* na reconstrução do campo.

A última etapa, consiste na reconstrução do campo vetorial que nada mais é que uma versão suave do campo vetorial resultante da interpolação das escalas do espaço de escala (ver Figura 4.4(a)). As singularidades selecionadas pelo usuário fornecem amostras da função de escala  $s(x, y)$  no domínio. Para



4.3(a): Espaço de escala

4.3(b): Mudanças à reverter

Figura 4.3: A interface mostra para o usuário as mudanças topológicas em escalas próximas, aqui da 5 à 15 4.3(a) onde o usuário seleciona que mudanças ele deseja reverter (em roxo) 4.3(b).

reconstruir campo vetorial, precisamos interpolar essas amostras. Denotando por  $\bar{\mathbf{v}}_{i,j}(s)$  as amostras do campo vetorial na escala  $s$ , definimos o campo vetorial reconstruído  $\tilde{\mathbf{v}}$  no ponto da grade  $(x_i, y_j)$  por:

$$\tilde{\mathbf{v}}_{i,j} = \bar{\mathbf{v}}_{i,j}(s(x_i, y_j)) .$$

Qualquer método de interpolação pode ser utilizado, porém mudanças bruscas no parâmetro de escala podem comprometer a qualidade do resultado. Nesse trabalho, utilizamos dois métodos para a interpolação: funções de base radial (RBF) com base Gaussiana, e o método de interpolação de Shepard com núcleo Gaussiano (4). Ambos apresentaram resultado satisfatório (ver Figura 4.4(b)).

A interpolação com RBF de  $s(x, y)$  das escalas provenientes das singularidades selecionadas pelo usuário  $s_k$  em  $(x_k, y_k)$  é obtida por uma minimização por mínimos quadrados nos coeficientes  $\alpha_k$  de

$$\min_{\{\alpha_k\}} \sum_k \|s_{rbf}(x_k, y_k) - s_k\|^2 , \quad \text{onde} \tag{4-1}$$

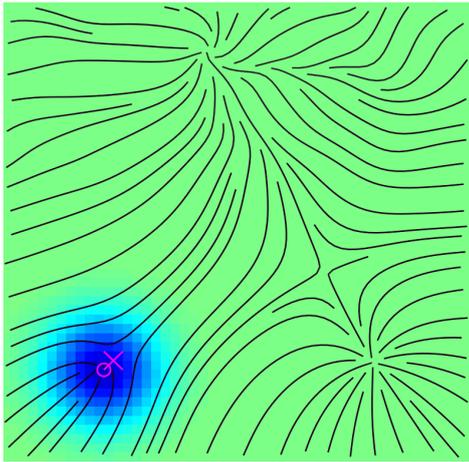
$$s_{rbf}(x, y) = \sum_k \alpha_k G_\sigma(x - x_k, y - y_k) . \tag{4-2}$$

O método de Shepard com núcleos (4) modifica a interpolação original

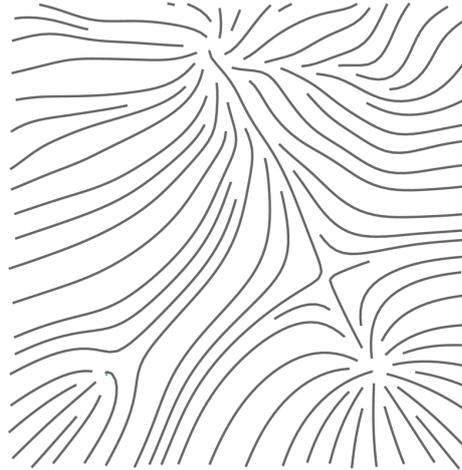
de Shepard (24) substituindo a distância Euclidiana por um núcleo:

$$s_{ks}(x, y) = \frac{1}{\sum_k G_\sigma(x-x_k, y-y_k)} \cdot \sum_k G_\sigma(x-x_k, y-y_k) \cdot s_k .$$

Uma propriedade importante nesse método é que a imagem fica limitada por  $[\min_k s_k, \max_k s_k]$ .



4.4(a): Função interpoladora



4.4(b): Campo vetorial reconstruído

Figura 4.4: Interpolação das escalas indicadas pelo usuário por uma função suave 4.4(a) que define a reconstrução do campo vetorial 4.4(b).

## 4.2 Segmentação do Campo Vetorial

Após a edição do campo vetorial, podemos fazer a segmentação do mesmo. A segmentação consiste em separarmos o domínio em grupos, de forma que linhas de fluxo com comportamento semelhante fiquem em um único grupo. Dizemos que as linhas de fluxo  $s_1$  e  $s_2$  têm o *comportamento semelhante* se  $s_1$  e  $s_2$  se originam em uma mesma singularidade e convergem para uma mesma singularidade.

Assim, após a detecção das singularidades e a construção das linhas de fluxo, resta detectar como o domínio pode ser particionado. Embora tenhamos adotado um método para a geração de linhas de fluxo igualmente espaçadas, nesta etapa, as estendemos até alcançarem o bordo ou até alcançarem uma singularidade. Para cada linha de fluxo então, associamos uma singularidade ao começo da linha e outra ao fim, podendo estas serem bordo ou não. Com isso, geramos um particionamento inicial.

Num primeiro tempo, poderíamos tratar o bordo inteiro como uma única singularidade. Na Figura 4.5, por exemplo, a segmentação ocorreu de maneira

natural, resultando em três grupos distintos: o primeiro formado pelo bordo, pela singularidade 1 e pelas linhas de fluxo que os conectam (cor azul); o segundo formado pelas singularidades 1 e 2 e pelas linhas de fluxo que as conectam (cor laranja); e um terceiro formado pela singularidade 2 e pelo bordo e pelas linhas de fluxo que os conectam (cor roxa).

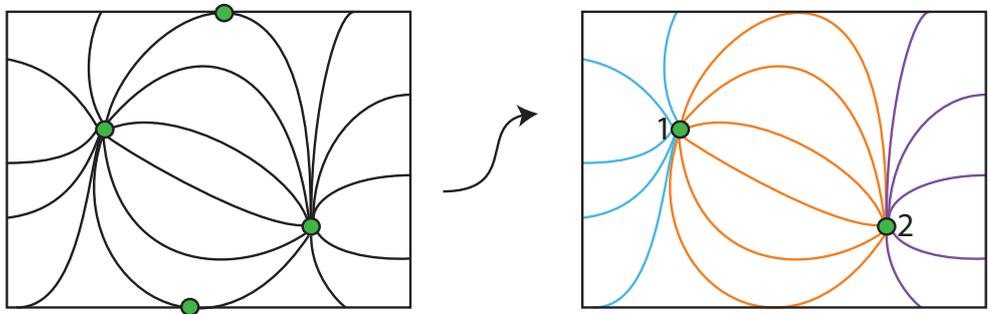
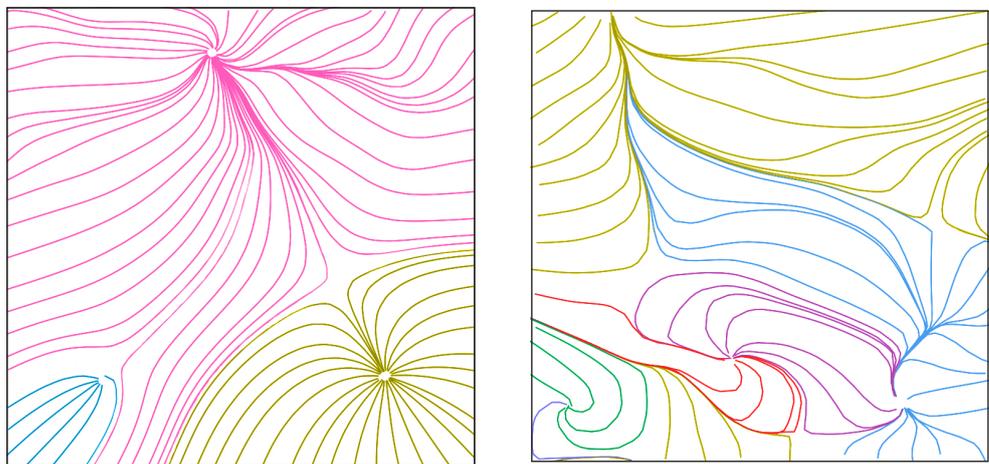


Figura 4.5: Segmentação do campo vetorial tratando o bordo como uma única singularidade.

Mas, nem sempre essa segmentação ocorre de forma tão natural. Seguindo a metodologia de tratar o bordo como uma única singularidade, não conseguimos segmentar, por exemplo, os campos da Figura 4.6 de maneira adequada, em particular, a separatriz saindo no canto abaixo a esquerda da Figura 4.6(a) não foi representada corretamente.



4.6(a): Campo segmentado em três grupos      4.6(b): Campo segmentado em seis grupos

Figura 4.6: Exemplos de campos segmentados tratando o bordo como uma única singularidade.

Em testes realizados neste trabalho, constatamos que nem sempre podemos utilizar o critério das separatrizes como solução. Utilizando a técnica de gerar linhas de fluxo igualmente espaçadas, a semente que irá gerar a separa-

triz nem sempre é válida, ou seja, a validade da semente ficará condicionada à densidade das linhas de fluxo no campo.

Precisamos então realizar um tratamento no bordo de forma que a segmentação ocorra de maneira adequada. O tratamento é realizado em duas etapas.

Como nossos dados estão armazenados em uma grade regular, podemos associar a cada célula de bordo, as linhas de fluxo incidentes e seu respectivo grupo. Tomemos a Figura 4.7 como exemplo para explicarmos como é feita a primeira parte do tratamento de bordo: o bordo desse campo ficará de acordo com a Figura 4.7(centro):

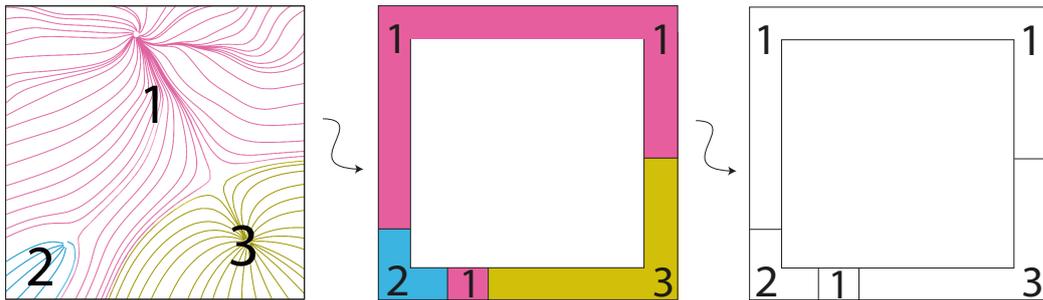


Figura 4.7: Esquema de campos segmentados tratando o bordo como uma única singularidade.

O que precisamos fazer na primeira parte é detectar quando algum grupo se divide em diferentes componentes conexas. Caso isso aconteça, particionamos o bordo criando uma nova singularidade por componente conexa (ver Figura 4.8).

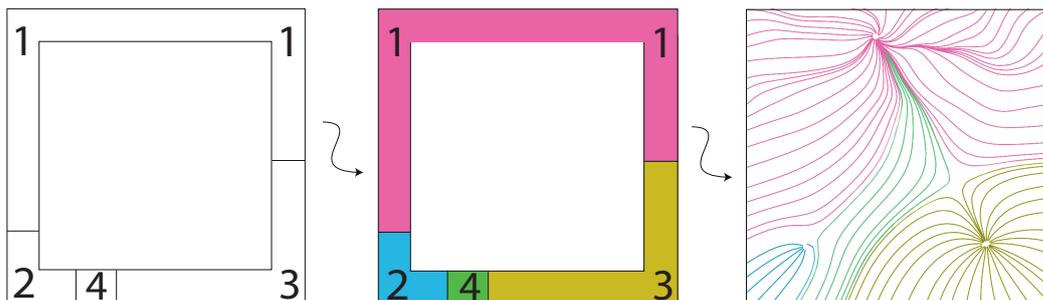


Figura 4.8: Esquema da primeira parte do tratamento de bordo.

A segunda parte do tratamento de bordo ocorre para o caso dos grupos onde as linhas de fluxo se iniciam e terminam no bordo, como ocorre na Figura 4.6(b). Aqui temos um caso onde podemos generalizar o tratamento de bordo.

Na Figura 4.9 observamos como seria a segmentação adequada realizada pelas separatrizes e na Figura 4.10, observamos a segmentação utilizando a primeira parte do tratamento de bordo.

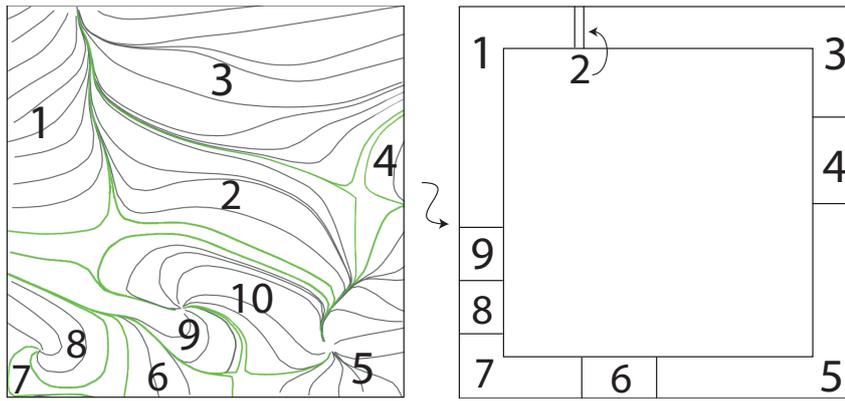


Figura 4.9: Segmentação a partir das separatrizes

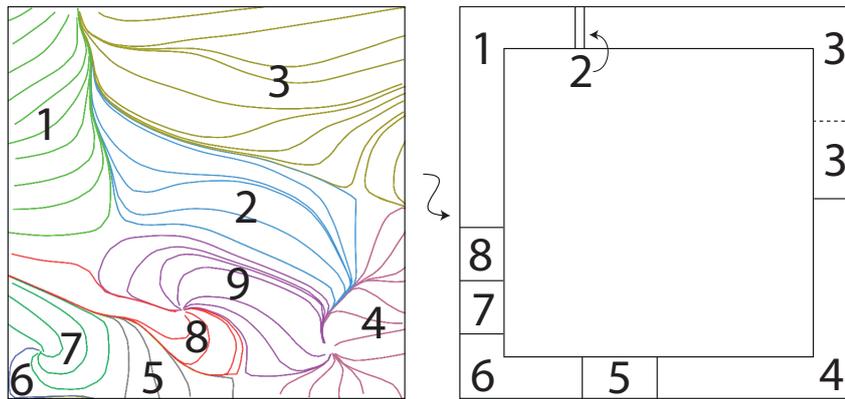


Figura 4.10: Segmentação com a primeira parte de tratamento de bordo

Como pode ser observado na Figura 4.9, existem grupos adjacentes, cujas linhas de fluxo se iniciam e terminam no bordo, mas que deveriam ser interpretados como grupos distintos - grupo 3 e 4, o que não ocorre na Figura 4.10 - grupo 3.

Para isso, podemos recuperar a informação de quais linhas de fluxo estão associadas às células de bordo. Como as linhas não se intersectam, basta descobrir quais linhas de fluxo delimitam os grupos percorrendo as células de bordo e selecionar as linhas de fluxo adequadas.

Na Figura 4.11, podemos observar a segmentação final do campo.

### 4.3 Imagens Auto-Animadas e Grafo Topológico

Embora a qualidade dos resultados obtidos por Chi *et al.* (6) seja alta, o custo para atingí-los também é relativamente grande, pois usa uma otimização força bruta com semente aleatória. Além dos dois parâmetros que são otimizados no posicionamento (cor inicial e tamanho do segmento), uma questão interessante é que a velocidade da convergência também depende da escolha das linhas de fluxo.

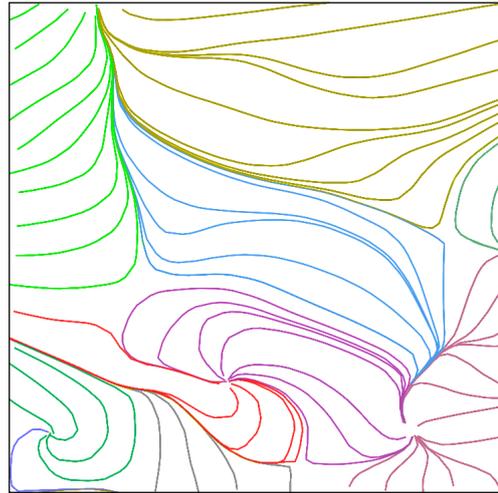
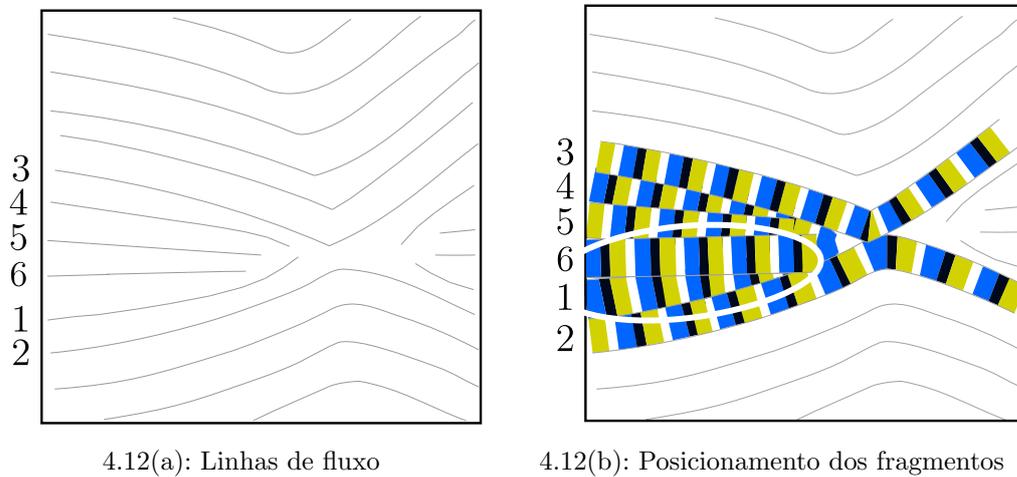


Figura 4.11: Segmentação final.



4.12(a): Linhas de fluxo

4.12(b): Posicionamento dos fragmentos

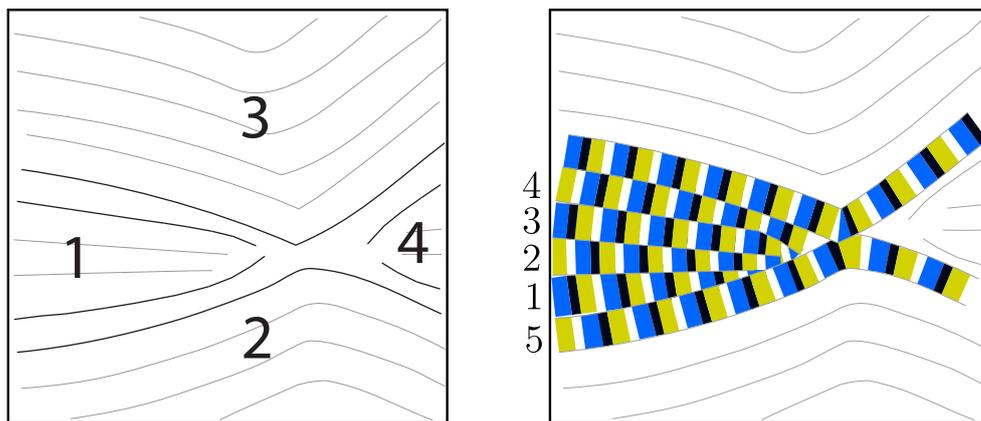
Figura 4.12: Posicionamento dos fragmentos desconsiderando a segmentação do campo.

Tomemos a Figura 4.12(a) como exemplo. Seja a linha de fluxo 1 a escolha inicial. O próximo passo, seguindo a otimização descrita na seção 3.4.2, é a otimizar em relação à sua vizinha. Suponha que a linha 2 seja a vizinha escolhida. Feito isso, faremos a otimização da linha 2 com a sua vizinha, linha 3, e da linha 3 com a 4, e assim seguiremos o processo. Quando desejarmos otimizar a linha 6 com a 1, o processo pode se complicar um pouco. Como pode ser observado na Figura 4.12(b), um padrão indesejado pode ser criado, o que prejudica o efeito da ilusão. Assim, será necessário recomeçar a otimização considerando novos parâmetros.

Um dos motivos desse problema ocorrer é que as linhas de fluxo escolhidas como vizinhas têm comportamento relativamente diferente, o que não evidencia qual seria o melhor parâmetro para a otimização.

Então, a otimização entre linhas com comportamento semelhante se da de maneira mais simples e robusta, permitindo uma estratégia adaptativa ao invés de força bruta. Uma forma de fazermos isso é justamente por meio da segmentação do campo descrita na seção 4.2, que particiona o campo vetorial em grupos de linhas de fluxo com comportamento semelhante.

Façamos então o processo de otimização no mesmo campo da Figura 4.12, mas agora utilizando a segmentação do campo 4.13.



4.13(a): Campo segmentado

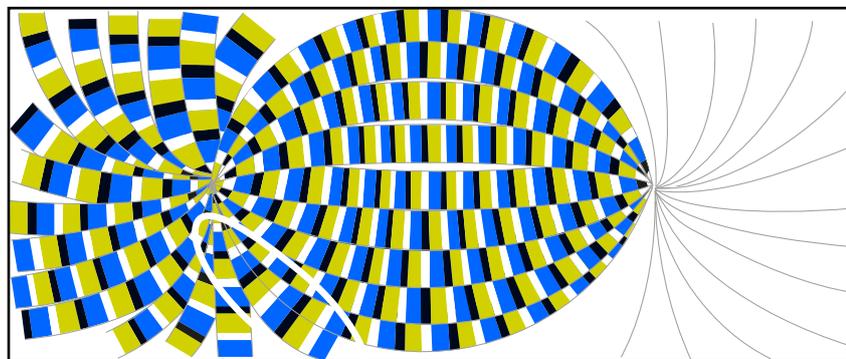
4.13(b): Posicionamento dos fragmentos utilizando a segmentação

Figura 4.13: Posicionamento dos fragmentos considerando a segmentação do campo.

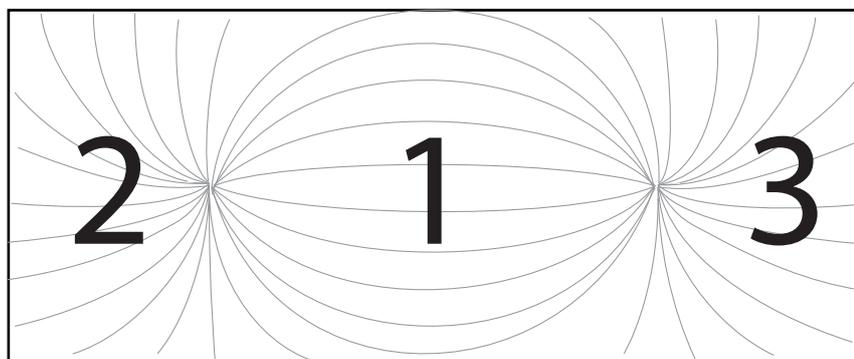
Como as linhas dentro do grupo se comportam de maneira semelhante, o primeiro parâmetro que modificaremos é a cor inicial das linhas, o que, em boa parte dos casos, já mostra um resultado satisfatório. Nas transições de grupos, otimizamos as linhas vizinhas entre grupo e continuamos a otimização agora no novo grupo.

Em alguns casos, a otimização por meio dos grupos pode não ocorrer de forma tão imediata, sendo necessário repetir o processo. A vantagem em utilizar os grupos, é que, diferentemente do que acontece quando é feita a otimização de forma global, não é necessário repetir o processo para todas as linhas do campo, será preciso apenas repetir o processo para o grupo corrente, deixando os demais intactos.

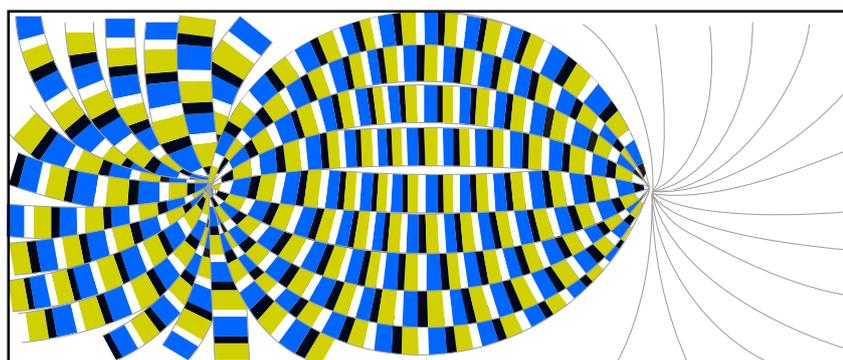
Um exemplo é mostrado na Figura 4.14. Nesta figura, primeiramente foi feita a otimização no grupo 1. Caso a transição entre o grupo 1 e 2 não seja razoável (Figura 4.14(a)), basta repetirmos o processo para o grupo 2 sem ser necessário modificarmos o grupo 1(ver Figura 4.14(c)). Caso a segmentação do campo vetorial não tenha sido considerada, não teremos como discernir quais linhas não precisarão ser modificadas, sendo necessário, portanto, repetir o processo para todas as linhas, aumentando o custo do processo.



4.14(a): Problema de posicionamento na transição dos grupos



4.14(b): Segmentação do campo



4.14(c): Reposicionamento dos fragmentos do grupo 2

Figura 4.14: Otimização considerando a segmentação do campo.